

Design • Build • Program • Learn • Compete

SERVO

FOR THE ROBOT INNOVATOR

www.servomagazine.com

MAGAZINE

March 2015

NOMAD

4WD Off-Road Bot
Chassis From
ACTOBOTICS

♦ **Robots And
Artificial
Intelligence**

♦ **Desktop Machine Tools**

Ideal For Hobby Robotics

♦ **Pascal On A PIC**

Full Featured, Highly Optimized, And FREE!

♦ **RoboGames Returns**

Rumors Of Its Demise Have Been Greatly Exaggerated!



MISSION MISSION POSSIBLE



Featuring an impressive high-def action camera and powerful brushless motors, the Q-Cop 450 is the performance-enhanced quadcopter you need for all your aerial missions. Its ergonomically-designed, multi-function transmitter, GPS guidance and remarkably stable characteristics provide amazing flight experiences up to 55 mph. Thrill in the excitement as you stream real-time Wi-Fi video. With its 120-degree field of view, 90-degree tilt control and shock isolation, the built-in camera will also capture your flights in stunning panoramic HD videos and 16-megapixel still photographs!

Additional Features:

- Autopilot with Return-to-Home
- Selectable GPS Directional Control
- 1600 ft. Radius Max. Range
- 22-33 mph Cruise, 45 mph Max. Speed
- 20-25 Minute Flight Times
- 3S 11.1V 5300mAh LiPo Battery with Charger
- 1080i/60fps, 1080p/30fps & 720p/60fps Video Formats
- Mobile Apps for iOS & Android

HITEC RCD USA, INC. | 12115 PAINE STREET | POWAY, CA 92064 | (858) 748-6948 | WWW.HITECRCD.COM |



SHERLINE

USA

Custom, Strong Robot Parts

Sherline Products manual and CNC Mills and Lathes offer the serious robot builder the power to build strong, repeatable custom robot parts from a variety of materials including nylon, delrin, aluminum, brass, titanium, steel and much more. Never again get stuck using inferior off-the-shelf parts or weak 3D printed plastic parts that are not ideal for your custom designed robot. With a little bit of practice you can be using one of our full CNC machines to turn out repeatable, perfectly finished parts. Our huge line of accessories gives you truly unlimited robot building possibilities.

sherline.com



Mention this ad and receive a **FREE** Tilting Angle Table with the purchase of any CNC machine package.

Visit us at

ROBOGAMES

3235 Executive Ridge • Vista, California 92081-8527, USA

Columns

07 Robytes

by Jeff Eckert

Stimulating Robot Tidbits

- Little Bot Helps Big Bot
- More than One Way to Skin a Chicken
- Security Bot Sports Vast Sensor Array
- No Tipping Required
- Own a Hollywood Actor



10 Ask Mr. Roboto

by Dennis Clark

Your Problems Solved Here

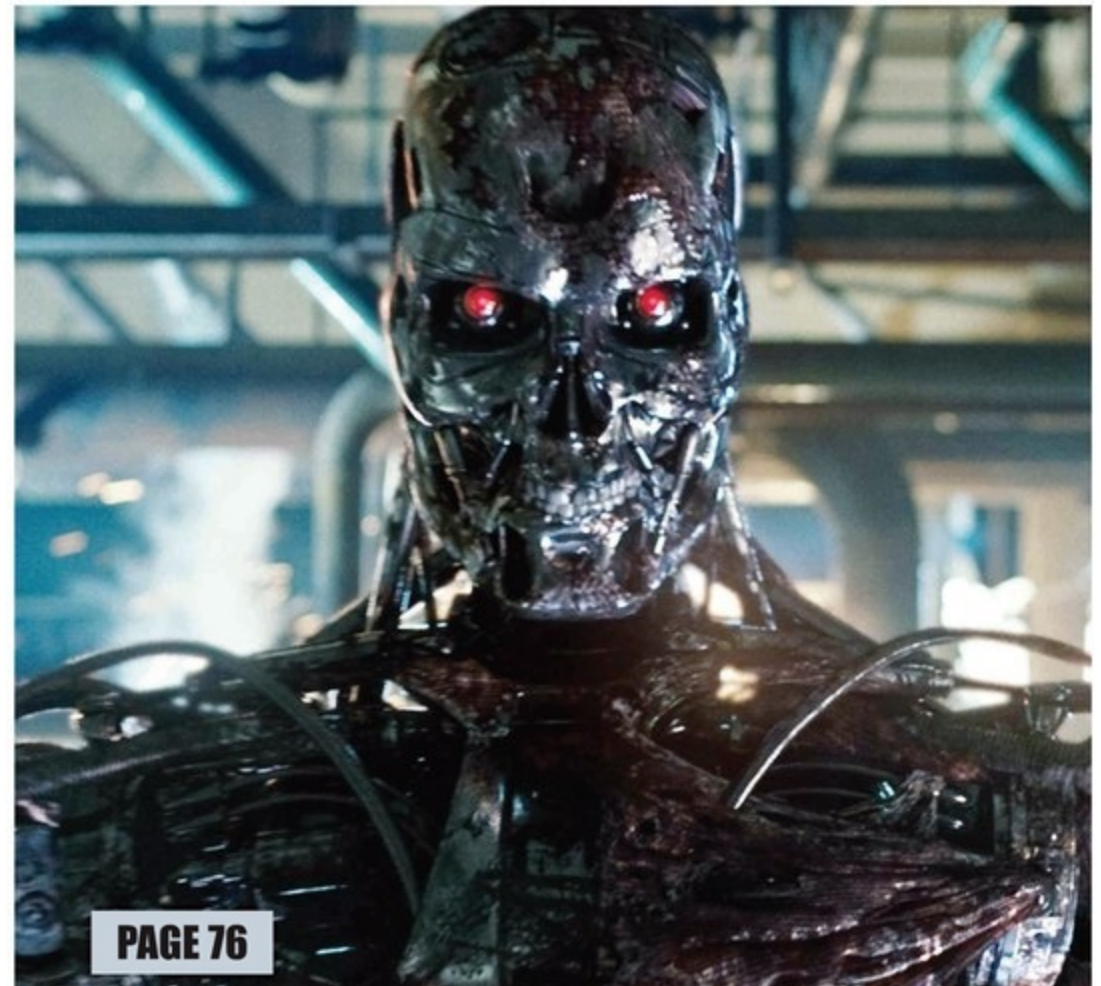
Discover the steps to get your bootloader back, plus how to deal with side loads on gearmotors.

76 Then and Now

by Tom Carroll

Robots and Artificial Intelligence

Universities, companies, plus robot and computer experimenters are all trying very hard to make AI a reality for robots. However, many experts have a great fear of these advances in artificial intelligence.



75 MaxRoboTech Comics

Either Way, You Do Not Lose

Departments

06 Mind/Iron

Mapping Kinetic Art to Robotic Science

14 New Products

17 Events Calendar

21 Showcase

66 SERVO Webstore

82 Robo-Links

82 Advertiser's Index

18 Bots in Brief

- Robots Enter the Ice Age
- (Cre)Adapt and Overcome
- Bat Bot to the Rescue
- Cooking Up Knowledge from YouTube
- Printed Actuators, Actually
- Phone for Brains
- Drawing a Line in the Sand



32 The Robot You've Always Wanted

by John Blankenship and Samuel Mishal

Part 3 explains how to program navigation behaviors, so your robot can make it from Point A to Point B without crashing.

38 Using Desktop Machine Tools to Build Your Next Robot

by Gordon McComb

Historically, mills and lathes have been leviathan in size, making it unrealistic for hobbyists to have them in their workshops. However, desktop machine tools now make it possible to build more elaborate robots at home. Discover what's available and what you can do with it.

42 Unwrapping a daVinci 3D Printer

by Steven Nelson

Robotics is well known for its camaraderie among competitors. Here's a story of how that friendship led to the acquisition of a 3D printer.

50 CGTrader's 3D Robot Challenge

Many unique contests are popping up these days with all the AI grumblings in the spotlight and all the new cool tools currently available. Maybe this one can inspire you.

53 RoboGames Returns!

by Ray Billings

As the largest robotic event in the world, participants and spectators alike were sorely disappointed when RoboGames did not take place in 2014. This year is gonna be way better with the return of this popular event.

56 Actobotics' Nomad

by Jeff Cicolani

Looking for an off-road robot chassis? Check out this review to see if Nomad's four-wheeled frame is the right platform for your next build.

62 Make a MiP Hack Back

by Dave Prochnow

In late 2014, WowWee Group Limited came out with another commercial robot success: MiP. So, of course, the first thing we want to do is hack it.

70 Getting Started with Pascal on a PIC

by Thomas Henry

Want to try advanced algorithms on your PIC-based robot? Pascal for the PIC provides a highly structured tool for your next robotics experiment.



The Combat Zone

22 A Veteran Builder Experiments with Brushless Drives

25 BUILD REPORT:
The Evolution of Slim Pickens

27 PARTS IS PARTS:
Product Review: AndyMark 7075 Aluminum Sprockets

29 Testing a Theory ...

Mapping Kinetic Art to Robotic Science

The process of mapping a cognitive field to AI rules is relatively straightforward. In one paradigm, a knowledge engineer maps a specific domain — say, navigation — into rules that can be executed under the proper conditions to arrive at an answer. Another approach is to feed every condition and outcome into a neural network, and train the system to recognize and respond to specific patterns. There's no guarantee either approach will work in a given situation, but there are established processes for training, writing rules, and testing for success.

Mapping a kinetic art — say, honing a blade on a cuticle or whetstone — to robotics is often an entirely different matter. Continuing with the honing example, human experts rely on visual changes in the slurry, as well as the “feel” of the blade as it passes over the honing stone to determine what to do next. There isn't a simple cookbook process involved — hence, the reference to kinetic art.

Because there's no simple mapping of art to science or process, the workaround is often to rely on some completely different method — one that relies on a specific known process — of achieving similar results. For example, instead of using a whetstone, sharpening of a blade can be achieved with a belt grinder in which the belt is impregnated with an abrasive. Instead of monitoring the “feel” of the blade as it passes over the whetstone, the task becomes determining the pressure and time required on the belt sander to achieve a given sharpness.

Blade connoisseurs know the difference between a blade sharpened by hand on a whetstone and one treated by a belt traveling tens of feet per second. They know that the belt may result in overheating and damage to the blade, and that more metal will be removed compared with more gentle hand sharpening on a whetstone. As a result, I'm sure there's a market for automated whetstone sharpening.

However, I've not seen such a device, and doubt that a commercially viable one could be developed — at least not yet. I can envision a robotic arm with enough precision, tactile feedback, and with a light enough “touch” to hone a razor — but such an arm would be cost-prohibitive today. For the time being, automating the mechanized belt grinder approach is probably the best commercial alternative to hand-honing. It's far from art, but it's a necessary detour until the science of kinetic mapping matures.

Given funding, it's likely that the first successes in mapping kinetic art to robotic science are likely to be in the fields of medicine or the military. Commercialization — and eventually the availability to our experimental robotics community — will inevitably follow. **SV**

FOR THE
ROBOT
INNOVATOR

SERVO
MAGAZINE

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

**ASSOCIATE PUBLISHER/
ADVERTISING SALES**
Robin Lemieux
robin@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

VP of OPERATIONS
Vern Graner
vern@servomagazine.com

CONTRIBUTING EDITORS
Tom Carroll
Dennis Clark
John Blankenship
Samuel Mishal
Dave Prochnow
Ray Billings
Thomas Henry
Michael Jeffries
Jeff Eckert
Kevin Berry
R. Steven Rainwater
Steven Nelson
Gordon McComb
Nate Franklin
Jeffrey Cicolani
Pete Smith
Hal Rucker
Jenn Eckert

CIRCULATION DEPARTMENT
subscribe@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

WEBSTORE MARKETING
Brian Kirkpatrick
sales@servomagazine.com

WEBSTORE MANAGER
Sean Lemieux

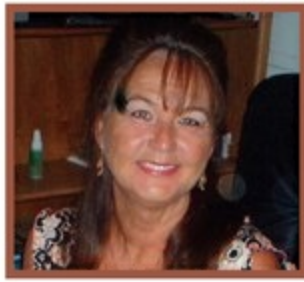
ADMINISTRATIVE STAFF
Debbie Stauffacher
Re Gandara

Copyright 2015 by
T & L Publications, Inc.
All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *SERVO Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *SERVO*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *SERVO*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: **430 Princeland Court, Corona, CA 92879.**

Printed in the USA on SFI & FSC stock.





Robytes

by Jeff and Jenn Eckert

Little Bot Helps Big Bot

In an interesting cooperative effort, a Kaman (www.kaman.com) K-Max unmanned helicopter and a Lockheed Martin (www.lockheedmartin.com) Indago quadrotor have teamed up. The K-Max — a medium-lift helicopter designed for lifting operations such as logging and construction — would appear to be an unlikely partner for the little 5 lb (2.27 kg) VTOL device, but the match seems to be perfect for firefighting operations. The Indago provides hover, perch, and stare capabilities that precisely identify and locate hot spots. It then relays the information to the K-Max which extinguishes the flames by dousing them with water.

In a test run, the K-Max autonomously dipped water from a pond and dumped more than 24,000 lb (almost 2,900 gal) of it onto the fire within an hour's time. "The unmanned K-MAX and Indago aircraft can work to fight fires day and night, in all weather, reaching dangerous areas without risking a life," noted Dan Spoor, VP of Aviation and Unmanned Systems at Lockheed.

An unmanned K-Max helicopter douses a fire under direction of an Indago quadrotor.



More than One Way to Skin a Chicken

Trondheim, Norway is home to Scandinavia's largest independent research organization, SINTEF (www.sintef.no). One of SINTEF's many focus areas is making Norwegian food production more profitable, efficient, and environmentally friendly. "Our aim is to automate absolutely everything we can think of on the food production line," revealed researcher Ekrem Misimi.

The group's recent innovations include a salmon filleting machine that uses X-rays to locate bones and a water jet to precisely clean the fish, thereby minimizing waste. However, the latest breakthrough is "Gribbot" — a robot that fully automates the process of removing breast fillets from chickens. The name comes from its resemblance to

a vulture's beak; "gribb" is the Norwegian word for "vulture." (This might seem unappetizing to us, but keep in mind that Norwegians actually eat lutefisk — a thoroughly disgusting dish consisting of whitefish treated with caustic lye soda.)

The keys to Gribbot's success are its 3D vision (based on a camera from a Kinect 2) and specially developed grippers that avoid spoiling the breast with marks or other defects.

"The fact is that almost half the food currently produced never reaches the consumer because it is lost along the production line due to the fact that we don't have the technology to process the raw materials," according to Misimi. "Automating this work will speed up production and make it more efficient."



The Gribbot team and their chicken bot.



Killing the Knightscope K5 security bot appears to be both fun and easy.

Security Bot Sports Vast Sensor Array

The concept of using robots for security and law enforcement isn't new, but the K5 Autonomous Data Machine from Knightscope (www.knightscope.com) is one of the latest entries to the field. According to the company, "Knightscope offers the only security solution capable of analyzing historical crime data, real time on-site data, and social media feeds to generate truly valuable crime predictions."

As the name implies, the K5 is intended to collect and analyze data rather than provide any active resistance to perpetrators, but it does gather a wide range of information using a GPS locator, LIDAR 3D mapping, 360° video, and a thermal imaging camera. It also sports night vision and can record audio; its capabilities include biological, chemical, and radiation detection. Another feature (for some reason) is optical character recognition. The K5 doesn't appear to carry any sort of weaponry, but unauthorized attempts to tamper with or kidnap it will result in an "ear-piercing audible alarm at the machine" and a red alert back at headquarters.

Knightscope's business model is based on a "machine-as-a-service" implementation, which basically means that you rent the K5 rather than buying it. The "introductory" (hmmm) rate is \$6.25 per hour, which sounds pretty cheap. It operates 24/7/365, so we're looking at \$54,750 per year. The average human security guard makes only \$28,351 (per salary.com), so maybe this isn't such a deal after all.

No Tipping Required

In days of yore, if you aspired to be a professional elevator operator, bowling alley pinsetter, or switchboard operator, your future seemed assured. Over the years, however, technology has eliminated those jobs. Likewise, if you plan to make a career of waiting tables in a restaurant, you may be looking down a dead-end street — at least if Infinium Robotics (www.infiniumrobotics.com) has anything to say about it. The Singaporean company is now serving up its Infinium-Serve autonomous flying robot servers, designed for restaurants and other food and beverage outlets.

Infinium notes that the deployment of multiple UAVs in confined spaces has been tough because the lack of space and tight margin-of-error requirements make collision avoidance difficult. However, the company has developed its own proprietary flight controller, as well as trajectory planning and model predictive control (MPC) algorithms that it claims produce positioning precision with no more than 1 cm error.

According to Infinium, the UAVs are simple to deploy (plug-and-play), can sense and avoid obstacles, and adapt to any restaurant's unique layout. Plus, they eliminate worker strikes and salary increases. Sounds good, but let's face it — these things aren't going to replace Hooters waitresses.



An Infinium-Serve UAV delivers food to the table.

Own a Hollywood Actor

If you already have more money than the crown prince of Abu Dhabi and are looking for a conversation piece, consider the Celebrity Robotic Avatar available from Hammacher Schlemmer (www.hammacher.com). Granted, it's pretty retro in appearance and its voice sounds like a one inch speaker at the bottom of a tin can. Plus, it isn't even truly autonomous, being controlled via an "intuitive wireless remote that is small enough to evade detection."

It can do astonishing things, however, like rolling forward and backward and turning around at three different speeds. There's also LEDs that flash in its mouth when it talks, "giving his speech a more natural quality." Pretty much the same way LEDs flash in your own mouth when you talk.

The 600 pound/72 inch machine can operate for three to four hours on a battery charge, or you can plug it into the wall for continuous use.

Well, okay, maybe it isn't all that amazing. However, units have appeared in movies (including Rocky IV), on television, and at numerous corporate functions. In fact, this is the only robot to have been admitted to the Screen Actors Guild. At a mere \$345,000, how can you resist? **SV**



*The Celebrity Robotic Avatar —
official member of the Screen Actors Guild.*

◆ Inertial Sensors ◆ Orientation & Tilt Sensors ◆ GPS-Aided Attitude & Heading Systems ◆



Best-in-class performance
at
Unbeatable prices

Our new UM7 Orientation Sensor represents the state-of-the-art in low cost, light weight attitude and heading sensors. We've combined the latest and greatest MEMS technology with efficient in-house calibration processes, so that we can offer unrivaled performance without emptying your wallet.

Come and see how we are revolutionizing the inertial sensing industry!

www.CHRobotics.com

CHROBOTICS

www.CHRobotics.com

ASK MR. ROBOTO

by Dennis Clark

Our resident expert on all things robotic is merely an email away.

roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

It is the slow part of the year right now, when I begin to plan my “roboting” activities for the rest of the year. Mr. Roboto lives in the Rocky Mountains, so that is where I tend to look for activities. In June, there is the SparkFun AVC that is always fun to watch and be part of. It’s a much nicer experience in June than it was in “the good ol’ days” when they did it earlier in spring! There are the Maker Faires that have popped up in Loveland and Boulder, CO to go to, as well. And, of course, there is the Critter Crunch in October where I enter what is usually the only fully autonomous critter to compete against (and often get trashed by) the human remote-controlled machines. So much fun! All in all, this is a great time to be a robot hobbyist, robot hacker, or maker, or any other bit of fun where folks used to sneer at us and call us nerds. Nowadays, the term geek actually seems to have a sort of status or prestige. Go figure. Anyway, on with your questions.

Q I have a Digilent chipKIT Max32 board that I have been using by programming it directly with Microchip compilers, which came about because I didn’t know I was going to kill the bootloader. I would like to get the chipKIT bootloader back. I see the “Burn Bootloader” option in MPIDE’s Tools menu, but I can’t seem to make it work. How do I get the bootloader back? I want to use the “UECIDE” IDE you wrote about with my board. Any help you give will be appreciated.

— Robo-Tommy

A I have never tried to use that option in MPIDE. You are already using the tool that you need to put the bootloader back on your MAX32: MPLAB. You didn’t say if you were on a Window or OSX PC, so I won’t assume either and recommend the Microchip IDE (integrated development environment) that runs on both of those platforms: MPLAB X. You may already have MPLAB X if you have been developing using the Microchip IDE. If you don’t, follow the steps below to get and use MPLAB X to re-install your bootloader.

What You Need to Get

If you don’t have MPLAB X, get it here:

www.microchip.com/pagehandler/en-us/family/mplabx.

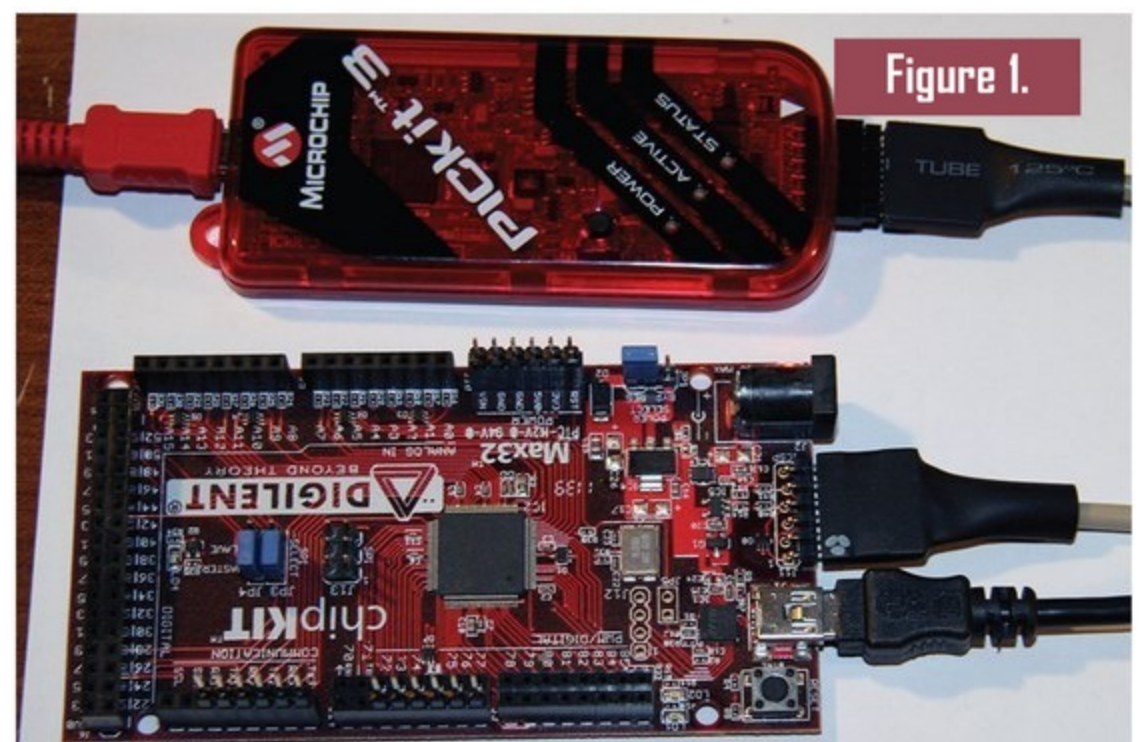
For MORE fun with other Microchip microcontrollers, add in the XC compilers. Pick the version for your OS. For this exercise, you only need this IDE.

Next, you need a way to program your MAX32 micro directly. Again, most likely you have this already, but I recommend (as the least expensive, but still very useful option) the Microchip PICKit™ 3 programmer. Refer to **Figure 1**.

Finally, you need the bootloader hex code file which Digilent provides right here: www.digilentinc.com/Products/Detail.cfm?NavPath=2,892,894&Prod=CHIPKIT-MAX32.

Scroll down the page until you see the link labeled “chipKIT™ bootloader image loaded into the MAX32’s PIC32 microcontroller at the factory.” Press Download to store this hex file on your computer; it comes as a zip file.

Unzip it now, so it is ready to use.



Post comments on this article at www.servomagazine.com/index.php/magazine/article/march2015_MrRoboto.

Install the IDE

When you download the IDE installer, it will come in one of two formats.

On OS X, it will be a ".dmg" disk image file. Double-click on that and a virtual disk volume will mount. Double-click on the installer image and just let it do its thing. The IDE executable will be in Applications>Microchip>mplabx.

On Windows PCs, the installer will be an ".exe" style installer. You know where the executable will be.

Creating the Bootloader Project

Here are the steps to create a project to burn the chipKIT bootloader to the MAX32.

Refer to the screenshots in **Figures 2** through **6**.

1. Start MPLAB X. Click on File->Import->Hex/Elf...Prebuilt file.
2. You will get a pop-up window. Use the "Browse..." button to select the hex file you downloaded earlier. This will name the project.
3. Click on the "Family" drop-down menu and select PIC32.
4. Click on the "Device" drop-down menu and select PIC32MX795F512L.
5. Select "PICKit3" as your Hardware tool.
6. The "Next >" button will now be enabled; click on it.
7. The pop-up will now show your project name (same as your hex file), locations, and folder. The project will be created in the same folder as your hex file. Click "Finish."
8. Your project is now in the left column; select it, then right-click on it and select "Set as Main Project" so that it is highlighted.
9. Connect your PICKit3 to a USB port if you have not done so already. Make sure that the MAX32 is powered through a USB connection to its micro-USB-B port; the PICKit3 can't power this board.
10. Connect the PICKit3 to the MAX32 as shown in **Figure 1**. The > on the PICKit3 denotes "pin 1." Make sure your connector cable connects to the pin on the MAX32 that is in the square hole, labeled "1" on the board. You will note (**Figure 6**) that the six-pin connector has staggered holes. This allows a Berg header to be "friction" fit into the holes to program the board and then be removed. Neat idea, Digilent!

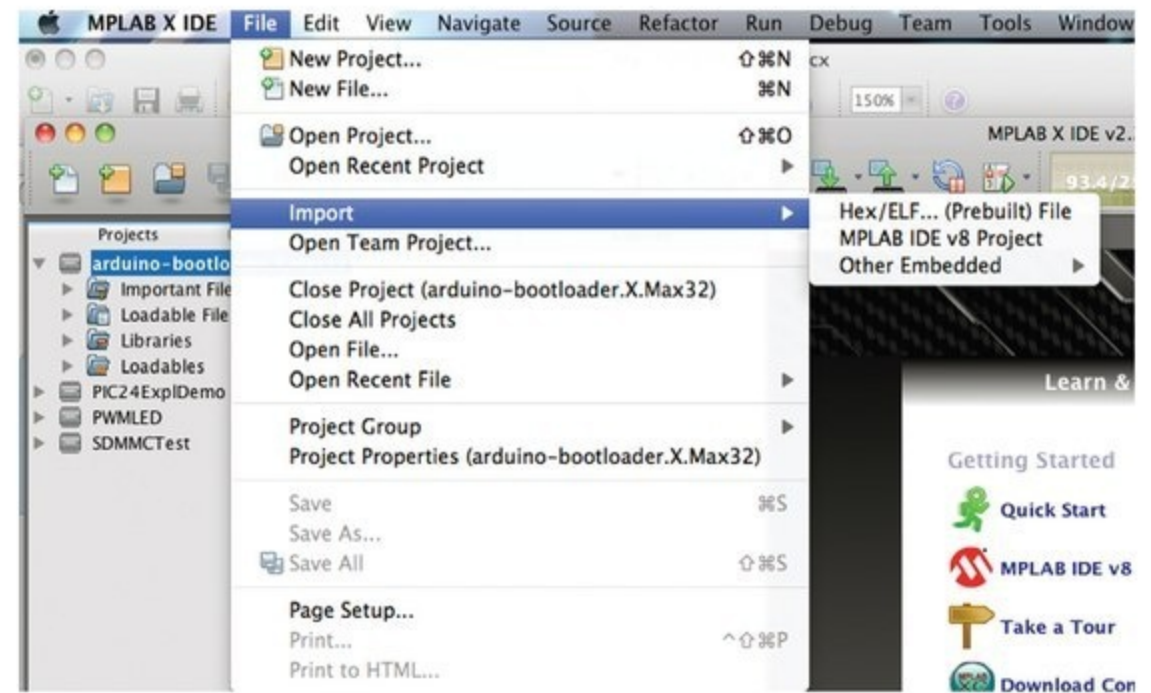


Figure 2.

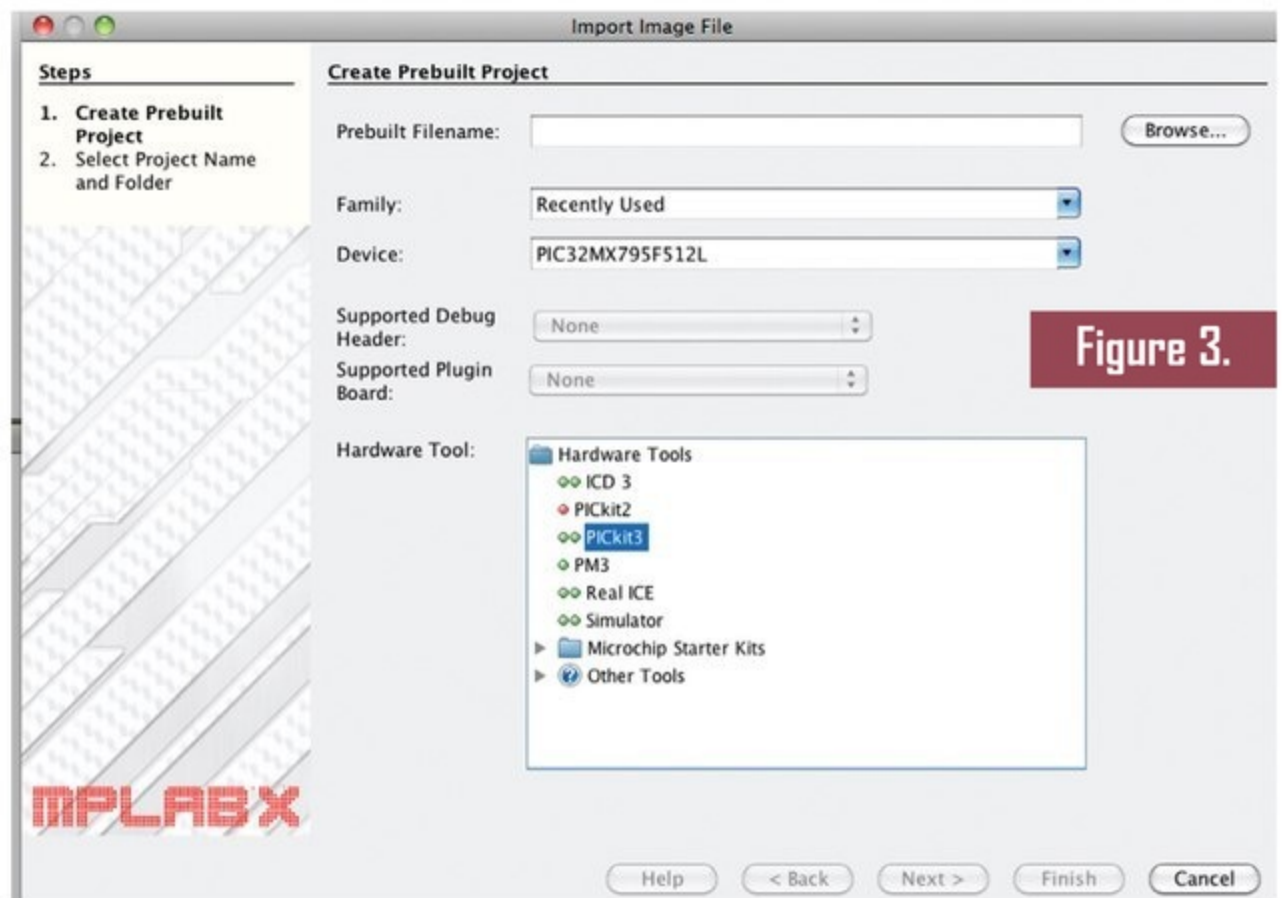


Figure 3.

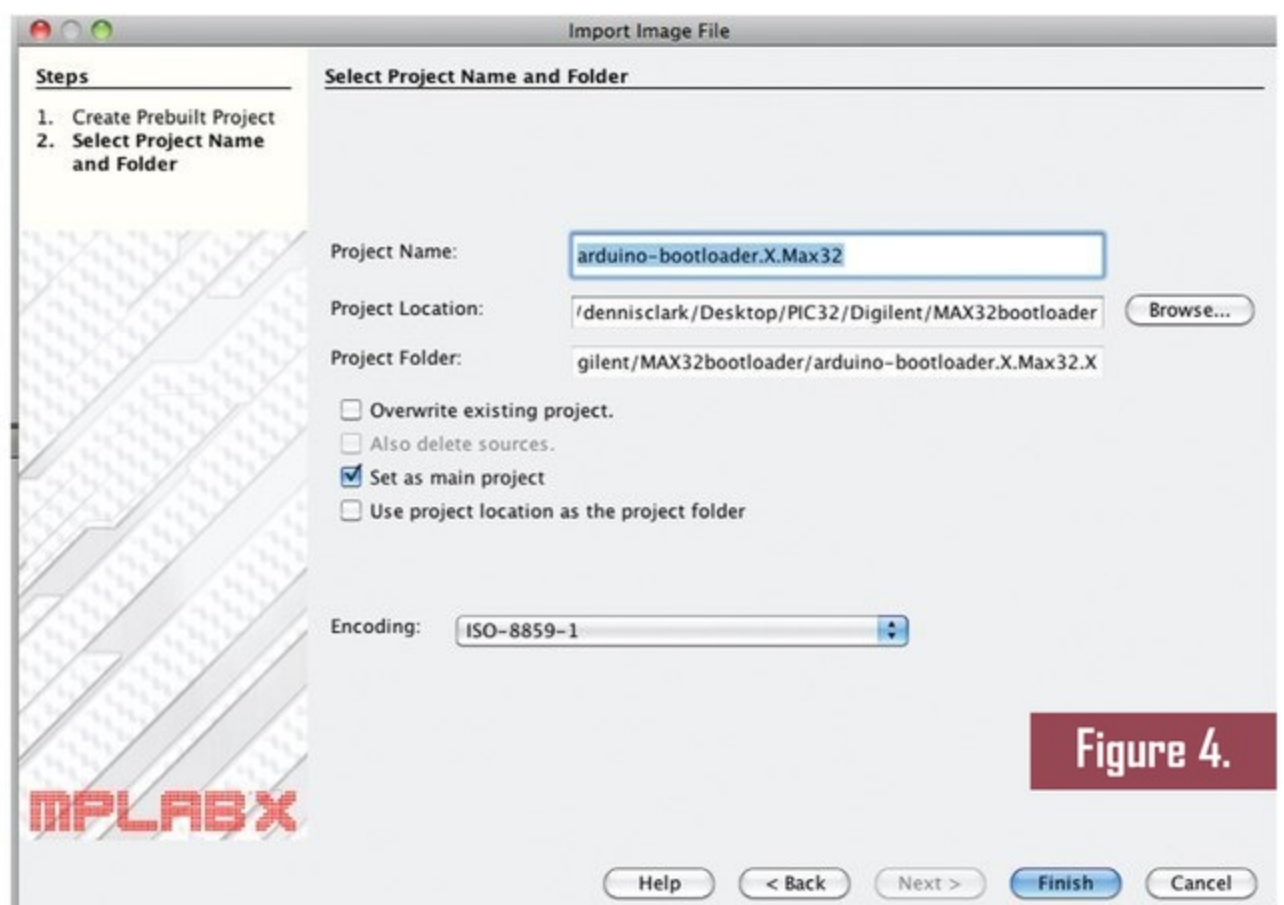


Figure 4.

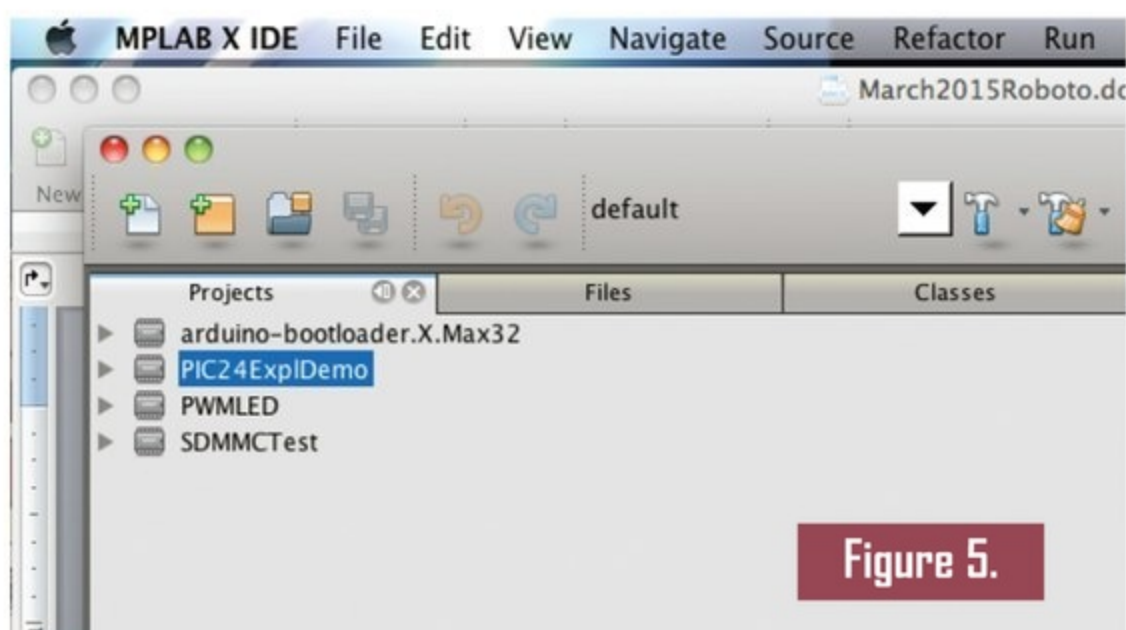


Figure 5.

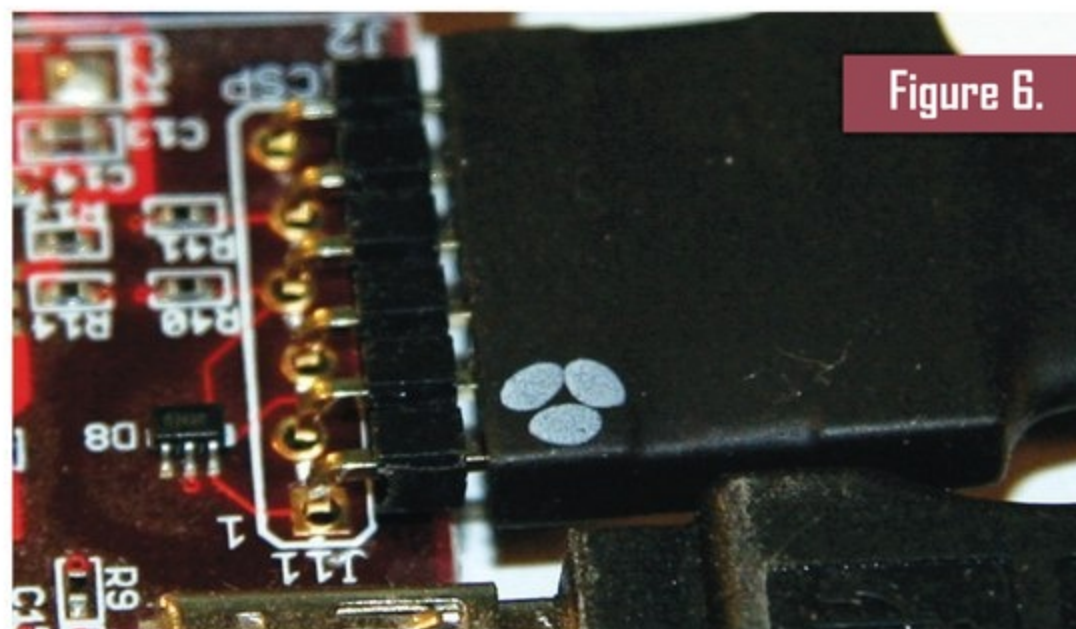


Figure 6.

Burning the Bootloader

Your project, board, and programmer are now all set to burn the bootloader.

Note the icon at the top of the window that looks like an arrow pointing down to a chip (**Figure 7**). If you hover your mouse over that icon, it will say "Make and Program Device Main Project." Click this, and you will program your board with the bootloader. Ta da!

Troubleshooting

If MPLAB gives you errors in the bottom window about not being able to connect to the board or not able to get the correct part ID, then check these two things:

1. Make sure that the six-pin Berg connector that is friction-fit into the programming connector on the MAX32 is very snug and doesn't wiggle much. This staggered pin layout is very clever. If you leave your Berg connector in there for too long, the pins will slowly bend to a new shape and your connection won't be good.
2. Right-click on your project name in the left column, then click on "Properties." You will see what's in **Figure 8**. Make sure that the SN: ... entry is selected. MPLAB X can run multiple sessions with multiple programmers/debuggers at the same time, so you have to explicitly select the one that a particular project is using.

You are now ready to go. Use your favorite chipKIT IDE, MPIDE, or UECIDE to create Arduino compatible programs on your MAX32 board.

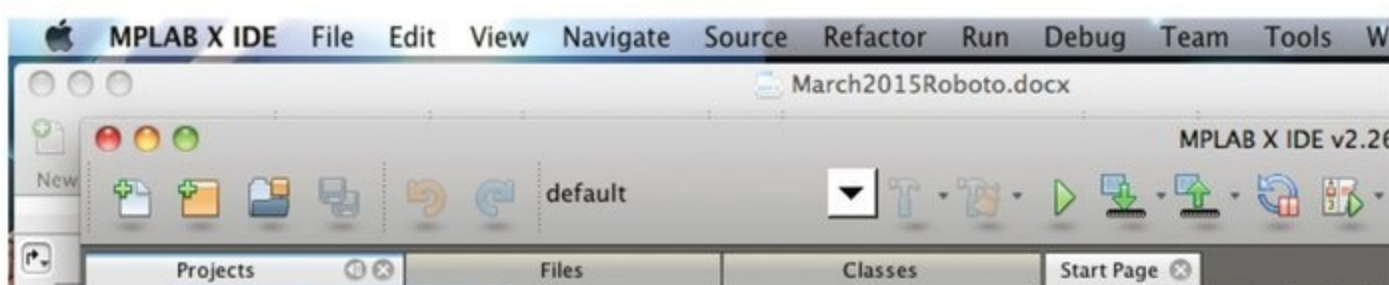


Figure 7.

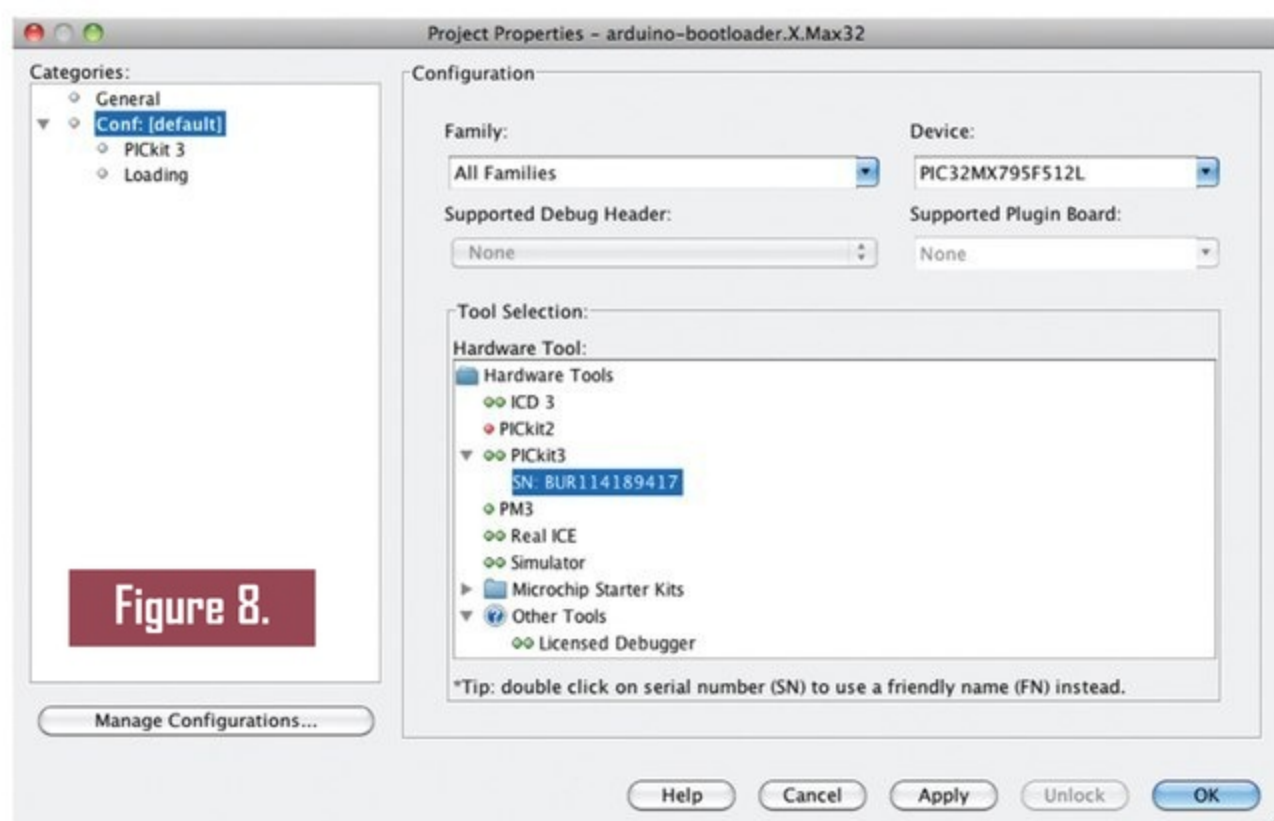


Figure 8.

As with many dedicated roboteers, each bot I build is bigger and more advanced than the previous one. My latest bot has a projected weight of 15 lbs. It has six drive wheels that support the chassis. Hence, each drive wheel will need to support 2.5 lbs. I plan on driving each wheel with a gearmotor attached to the frame and to a wheel. My question is: How much side load can the gearmotors reliably absorb? I have searched the 'net, looked at specification sheets, etc., and not found any data on this parameter.

I did a quick calculation on side load. From my spreadsheet, the present design will have about 8.3 lbs of side load on the gearbox at the bearings. If you adjust the various parameters on the spreadsheet, the side load can change significantly.

This concept applies to servos or any mechanism that drives a load that has forces

perpendicular to the shaft. Servos are another area where we tend to have side loads, and finding data on side loads for these has also been an exercise in futility.

Your comments will be most appreciated! Thanks for the support.

— Ken Hemmelman

A. You have picked up on something that has long been a thorn in the side of us working with motors that have unsupported shafts. What you describe is typically called *radial loading*. If you Google radial loading, you will find the information you are looking for, but probably not for your motors unless you get lucky.

Hobby servos are the worst for robot wheels because they are — quite frankly — not designed to deal with continual radial loading. They are designed for strictly linear motion and intermittent use. That said, we also torture our gearmotors the same way.

However, these gearbox motors have significantly stronger shafts and the bearings are designed to handle side loads commensurate with their torque ratings.

Again, these gearmotors are NOT designed to carry weight continually either. They are used to move print heads on printers, spin tape pick-up reels, move belts on linear actuators, etc. So, we're killing them too. Because we usually get our gearmotors from surplus houses or resellers who get them from surplus houses, we never get the datasheets that spell out their limitations.

I had three good motors (Pittman, SOHO, and Escap) that I tried to find these specs on. I could find a manufacturer's spec sheet on only one of them. And — you guessed it — radial loading specs were not there. Everything else you would want to know about that motor was there, but no dice on the radial loading limits.

With light robots, you can get away without shaft support for a long time, but eventually the bearing wears

and your wheel will get sloppy. Your robot is heavy enough that you will wear your motors out faster than a little 500 gram Sumo robot. I recommend that you support your shafts. Since they are probably not long enough to carry a wheel and reach through to a side rail, you will need to get shaft couplers to extend the motor shaft.

Then, have the shafts sit in either sintered bushings or actual ball bearings to avoid unneeded friction that will rob power from your motor. Check out any of the heavy robots in *SERVO's* "Combat Zone" column and you'll see that their wheels are supported on both sides (of the wheel) to eliminate your problem.

No matter how tough your motor is, the radial load on that shaft will eventually destroy its output bearings and give your motor a wobble, which will then hamper any attempts at precision control. If you are getting your motors new, you might be able to get the specs that you want if you Google for radial load *and* your motor part number.

Good luck! More than likely you will want to look for shaft couplers that match your motor output shaft so that you can support your shaft on both sides of the wheel.

Well, that's all for this month. I am feeling the need to enter the 21st century, and use cameras and vision of some sort on my newer robots. Towards that end, I've gotten a 4D Systems uCam II that has a high speed UART communications bus and very high level control over a pretty decent camera/lens combination.

I have Googled this and not found anyone that has done anything with these yet, so I may be trail blazing! Wish me luck and I'll report back what I am able to do with this nifty camera.

So, until we meet again, keep on building robots.

Send your questions to me at roboto@servomagazine.com. I'll do my best to answer them. **SV**

Firgelli
www.firgelli.com

LINEAR SERVOS



L12-R Linear Servo

- Direct replacement for regular rotary servos
- Standard 3 wire connectors
- Compatible with most R/C receivers
- 1-2ms PWM control signal, 6v power
- 1", 2" and 4" strokes
- 3-10 lbs. force range
- 1/4" to 1" per second speed ranges
- Compatible with VEX



L16 Linear Actuators

- 2", 4" and 6" strokes
- 10 - 40 lbs. force range
- 1/2" to 1" per second speed ranges
- Options include Limit Switches and Position Feedback

New!

PQ12 Linear Actuator

- Miniature Linear Motion Devices
- 6 or 12 volts, 3/4" stroke
- Up to 5 lbs. force
- Integrated position feedback or limit switches at end of stroke
- External position control available



Linear Actuator Controller (LAC)

- Will drive any Linear Actuator with position feedback
- Up to 24v and 4 Amps
- USB connectivity to drive the actuator with your computer
- Adjustable speed, limits and sensitivity



L12-NXT Linear Servo

- Designed for LEGO Mindstorms NXT®
- Plugs directly into your NXT Brick
- NXT-G Block available for download
- Can be used with Technic and PF
- Max. speed: 1/2" per sec.
- Pushes up to 5 lbs.
- 2" and 4" strokes



Available Now @
www.firgelli.com



NEW PRODUCTS



Nomad 4WD Off-road Robot Chassis

ServoCity is now offering the Nomad 4WD off-road chassis kit which is an easy to assemble robotic platform capable of going places a normal chassis can't. What makes this chassis kit different is not only its vast amount of attachment points for various add-ons, but also the fact that it is easily and fully configurable.

The chassis is equipped with four 5.4" diameter by 2.25" wide heavy duty tires, four super duty/ball-bearing planetary gearmotors with full metal gears, and a ball-bearing pivot suspension. Also included in the 4WD chassis kit is a large central ABS plastic body with two large access panels that open up to a cavity large enough for a 7.2V NiCAD/NiMH or other LiPo battery and electronics to fit comfortably. Its central body also has a multitude of 0.77" hub patterns and cutouts at the bottom to run motor wires through. Since it is made out of ABS plastic, drilling holes for additional mounting options is simple. Retail price is \$279.99.

Force Servo Arm

Also available from ServoCity is the Force servo, which is a new type of a servo drive for remotely controlled devices. The majority of modern servo drives

maintain a controlled position of the arm. Other servo drives are responsible for cyclic rotation with variable speed. Force Servo or F-Servo is a servo drive with controllable action force.

The value of force is proportionate to the control signal and does not depend on the position of the arm. A force sensor (dynamometer) is used for

controlling the force.

This arm simplifies many mechanical devices where it's necessary to control the action force created by a servo drive or the action force of a mechanism controlled by a servo drive. When used instead of a servo drive with a controlled position of the arm, the arm makes it possible to improve the technical characteristics of many devices and mechanisms.

The key areas of application are remotely controlled light drones, robots, radio-controlled models, and rotation devices for video cameras. Retail price is \$24.99 each.

For further information, please contact:

ServoCity

Website: www.servocity.com



One-step Design-in of 3D Gesture Recognition in Embedded Devices

Microchip Technology, Inc., has announced the second member of its award-winning and patented GestIC® family. The new MGC3030 3D gesture controller features simplified user-interface options focused on gesture detection, enabling true one-step design-in of 3D gesture recognition in consumer and embedded devices. Housed in an easy-to-manufacture SSOP28 package, the MGC3030

expands the use of 3D gesture control features into high volume cost-sensitive applications such as toys, audio, and lighting.

The simplicity of gesture-detection integration offered by the MGC3030 is also achieved through Microchip's free downloadable AUREA graphical user interface (GUI) and easily configurable general-purpose I/O ports that also allow for host MCU/processor-free usage. The MGC3030's on-chip 32-bit digital signal processor executes real time gesture processing, eliminating the need for external

cameras or controllers for host processing, and allowing for faster and more natural user interaction with devices.

Further simplifying the design process and accelerating time to market, the MGC3030 makes full use of the GestIC family development tools, for example, Microchip's Colibri Gesture Suite — an on-chip software library of sophisticated yet easy-to-use gesture features.

Intuitive and natural movements of the human hand are recognized, making the operation of a device functional, intuitive, and fun. Without the need to touch the device, features such as Flick Gestures, the Air Wheel, or the proximity detection perform commands such as changing audio tracks, adjusting volume control or backlighting, and many others. All gestures are processed on-chip, allowing manufacturers to realize powerful user interfaces with very low development effort.

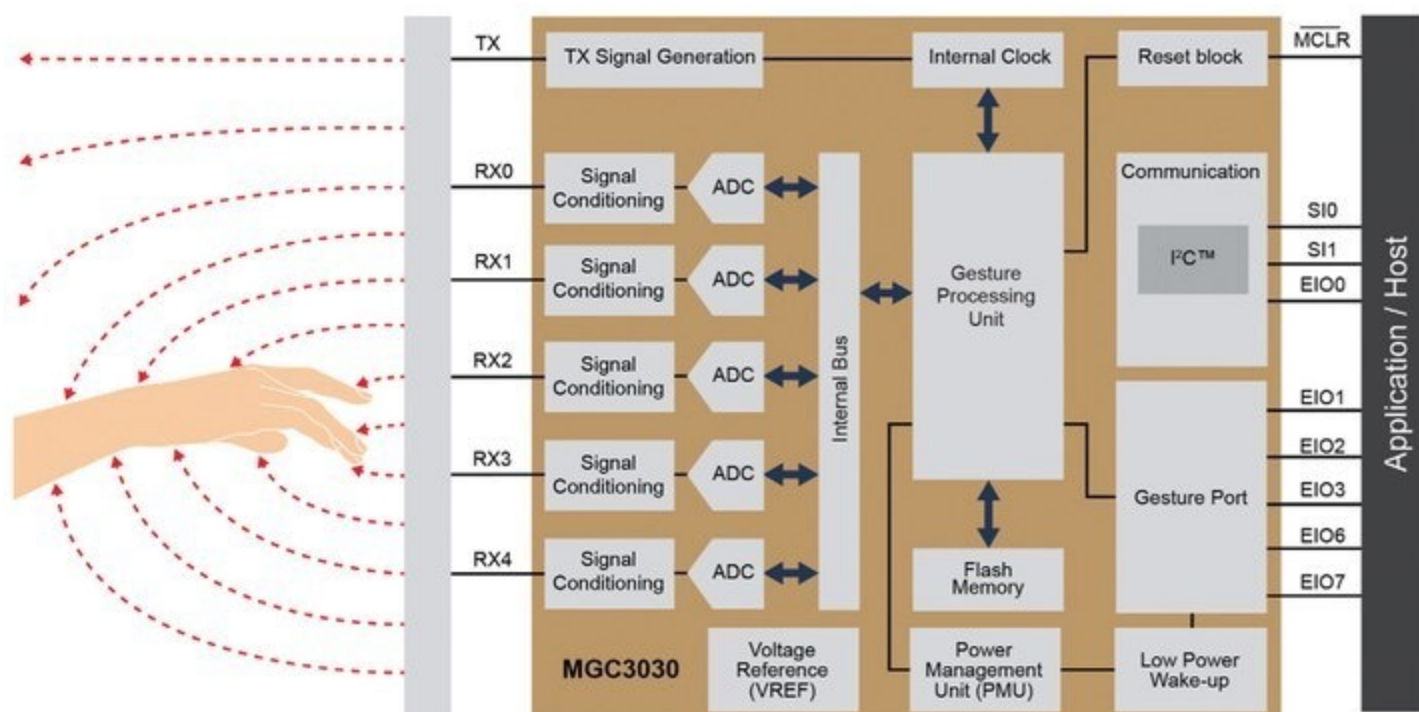
To enable development with the MGC3030, Microchip's Woodstar kit is available for \$139. The MGC3030 featuring GestIC technology is attainable in a 28-pin SSOP package and is priced at under \$2 each in high volumes.

Power MOSFET Drivers

Also from Microchip are the first power MOSFET drivers in a new product family: the MCP14A005X and MCP14A015X. These drivers feature an entirely new driver architecture for high speed operation. Available in SOT-23 and 2 x 2 mm DFN packages, the new devices are among the smallest packaged MOSFET drivers on the market today. The small packaging enables higher power densities and smaller solutions, while the design targets fast transitions and short delay times that allow for responsive circuit operation.

Additionally, these MOSFET drivers include low input threshold voltages that are compatible with low voltage MCUs and controllers, while still maintaining strong noise immunity and hysteresis.

The MCP14A005X and MCP14A015X MOSFET drivers low





Personal CNC Mills



Shown here with optional stand and accessories.

Shown below is an articulated humanoid robot leg, built by researchers at the Drexel Autonomous System Lab (DASL) with a Tormach PCNC 1100 milling machine. DASL researcher Roy Gross estimates that somewhere between 300 and 400 components for "HUBO+" has been machined on their PCNC 1100.



PCNC 1100 Series 3
starting at:
\$8480
(plus shipping)

www.tormach.com/servo



input threshold is compatible with various PIC microcontrollers (MCUs) and dsPIC digital signal controllers (DSCs), even when operating at lower voltages. This enables customers to design applications with MCUs operating as low as 2.0V, using the MOSFET driver to boost the output signals to 18V, reducing power loss in the controller, and minimizing conduction loss in the power MOSFET. These drivers are designed for use in consumer electronics markets, including embedded power conversion, brushed DC motor, unipolar stepper motor, and solenoid/relay/valve control applications, among others.

The MCP14A005X and MCP14A015X are available now for sampling, and volume production in SOT-23 and 2 x 2 mm DFN packages at prices ranging from \$0.50 to \$0.61 each in quantity.

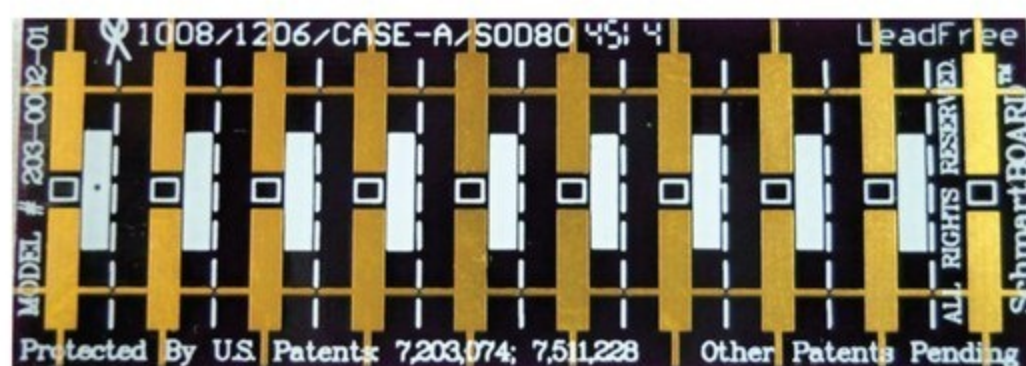
For further information, please contact:

**Microchip
Technology, Inc.**

Website: www.microchip.com

Products for PCB Repair

SchmartPatch — new from SchmartBoard — allows users to add surface-mount components to a substrate such



as a printed circuit board (PCB), or to replace existing components. SchmartPatch boards each support multiple pre-defined components and allow one to cut off a pad from the SchmartPatch (using scissors or a blade), stick the pad on to the surface of the substrate (utilizing adhesive already on the back of the SchmartPatch), and then solder a component onto the SchmartPatch.

The component pads are routed to other pads that allow one to solder jumper wires from the SchmartPatch to make the electrical connection to the substrate.

SchmartBoard has released six types of SchmartPatch boards. Each comes in packs of four for a retail price of \$15. Volume purchases can be arranged with SchmartBoard directly for larger rework needs.

For further information, please contact:

Schmartboard

Website: www.schmartboard.com

WHERE WILL YOU BE ON 05.04.15?

2015 will be the most important year for unmanned systems since our technology hit the land, air and sea – the year when technology, applications, policy and innovation converge to transform the future of our industry.

On 05.04.15, join the largest global community of commercial and defense leaders in unmanned systems, drones and intelligent robotics at Unmanned Systems 2015.



**AUVSI's
UNMANNED SYSTEMS
2015**

CONFERENCE: MAY 4-7 | TRADE SHOW: MAY 5-7

GEORGIA WORLD CONGRESS CENTER
ATLANTA | USA

**UNMANNED
INNOVATION
CONVERGES
05.04.15**

WWW.AUVSISHOW.ORG

MARCH

7-8 FIRST LEGO League of Central Europe
*Deutsches Museum
 München, Germany*
 Student teams compete in annual championship.
www.first-lego-league.org/en/fll/regions.html

7-8 METU Robotics Days
METU Culture & Convention Center, Turkey
 Events include Search & Rescue, Sumo, Multiple Mini Sumo, Minefield, Solve the Maze, Trash Hunter, Line Follower, and Free style.
http://odturobotgunleri.org.tr/tr

13-14 AMD Jerry Sanders Creative Design Contest
University of Illinois at Urbana-Champaign, IL
 Student teams compete in a different event each year. See website for details on this year's challenge.
http://jsdc.ec.illinois.edu

15-19 APEC Micromouse Contest
Charlotte, NC
 Speedy, tiny, autonomous robots compete against each other to solve a complex maze in the shortest possible time.
www.apec-conf.org

19-21 Festival de Robotique
*Montreal, Quebec
 Canada*
 Student competition with events for robots built by students in various age groups.
http://festivalde-robotique.ca

19-22 Techkriti RoboGames
*Kanpur, Uttar Pradesh
 India*
 University competition for robots built by student teams.
www.techkriti.org/#/competitions/robogames/

21 Manitoba Robot Games
*Winnipeg, Manitoba
 Canada*

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it. For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **<http://robots.net/rcfaq.html>**.

— R. Steven Rainwater

Events include Prairie Sumo tethered, Prairie Sumo autonomous, Mini Sumo tethered, Mini Sumo autonomous, Tracker Pull, Line Follower, Super Scramble, Robo Critter, and LEGO NTX/Mindstorm autonomous.
www.scmb.mb.ca

25-29 Apogee iStrike
Zuarinagar, Goa, India
 University students build robots to compete in events that include Robots at War and Full Throttle Grand Prix.
www.bits-apogee.org

27-29 Cognizance
IIT Roorkee, India
 Events this year include Armageddon, Powerdrift, Hephaestus, Poseidon, Breach Wars, Cyborg Breakin, and Robosapiens.
http://cognizance.org.in

AndyMark®
 Robot Parts Experts



Thrifty Throttle

Run motor controllers or servos with AndyMark's easy and affordable PWM signal generator.

- PWM Generation
 Standard servo range - 1000ms-2000ms
 Arduino extended range - ~540ms-2300ms
- Comfortable one-hand operation
- 9V Battery Operation (battery not included)

Visit **AndyMark.com** and view our wide selection of robot parts!

Use coupon code **"SERVO5"** for 5% off your next order

Kokomo, Indiana • www.AndyMark.com • 877-868-4770

bots IN BRIEF

ROBOTS ENTER THE ICE AGE



Dr Alex Forrest with the autonomous underwater vehicle UBC-Gavia on a research expedition to Antarctica in October 2014. A new AUV facility has been established at the Australian Maritime College in Launceston, Tasmania, to build next-generation hybrid autonomous vehicles capable of exploring hundreds of kilometers under meters-thick ice.

World-leading research at the Australian Maritime College (AMC; a specialist institute of the University of Tasmania) will allow the nation to capitalize on the robotic age of Antarctic exploration.

The \$24 million Antarctic Gateway Partnership launched by the Australian government in November 2014 includes \$7.5 million to establish a marine technology hub to build next-generation hybrid autonomous vehicles capable of exploring hundreds of kilometers under meters-thick ice.

Leading this research and development is AMC's autonomous underwater vehicle (AUV) and fluid mechanics expert, Dr. Alex Forrest.

"Tasmania has a real critical mass in terms of Antarctic research — robotic or otherwise — that we are capitalizing on through this project," Dr. Forrest said.

"There are four main areas of research in the Antarctic Gateway Partnership: ice shelf cavities, marine biology, solid earth, and marine technologies. We'll be leading the marine technologies theme and developing a new AUV that can go underneath ice cavities, sample the biology, and provide feedback for all the other themes."

Work is underway to establish the AUV facility at AMC in Launceston, including the recruitment of four new positions. The aim is to research, design, and build the robots

over the next three years in readiness for deployment in 2018 — the final year of the project.

"We'll be looking at what the interesting science questions are and how an AUV facility can develop multiple types of vehicles with the support staff necessary to target those projects," Dr. Forrest commented.

"One of the biggest challenges now is the conceptual design phase. We need to build a vehicle, we need to build it quickly; so, what instruments do we need on board? What are the attributes that we want the vehicle to have? In essence, these robots are relatively simple in design, but it's their range, capabilities, instruments, and data sensors that we want to expand upon in a novel way."

Dr. Forrest has spent nearly a decade specializing in under-ice deployments in the Arctic and Antarctic. He was part of an international research expedition to Antarctica in October 2014 where he studied algae growth underneath sea ice using the AUV UBC-Gavia — a vehicle based at AMC.

The project is a collaboration between AMC and the Institute for Marine and Antarctic Studies (University of Tasmania), the University of Canterbury (New Zealand), and Aarhus University (Denmark), with funding from the New Zealand Antarctic Research Institute.

It aimed to compare measurements taken above ice using traditional ice coring techniques with measurements made below the ice with the AUV. The seven-member team spent 21 days conducting their investigations at Cape Evans in the New Zealand-managed territory accessed from Scott Base.

"We're trying to understand the dynamics of algae growth and distribution because these organisms are the first step in the Southern Ocean ecosystem. So, in order to understand the productivity of this system in under-ice conditions, we need to understand the distribution of the ice algae," Dr. Forrest explained.

Robotic under-ice sampling is far more efficient and accurate than traditional methods. Divers are limited to operating within a 20 meter radius of the drilled hole and gathering core sample data from single locations. Robots can be used for broadscale measurements that capture any horizontal variability, rather than estimating it based on fixed point sampling.

"This AUV has very expensive navigation equipment that estimates its exact location, providing the longitude and latitude for every measurement down to centimeter accuracy. So, we know exactly where we are when we're taking a measurement, and from that we can build three-dimensional

(CRE)ADAPT AND OVERCOME

Jean-Baptiste Mouret's hexapod taught itself how to walk again after losing a leg. Damage resilience is certainly critical to being able to remain mobile while exploring that scary and dangerous world outside of a lab, but adaptation can be taken even farther. Mouret's new robot is exploring exactly how far that is on six adaptable wheels.

On flat ground, the Creadapt robot can use its wheels, but as soon as it encounters some other surface like dirt, grass, rocks, or rubble, it'll change its gait to whatever is most efficient. How does it know what the most efficient gait is for a given surface? It doesn't. Yet.

Fundamentally, a robot losing a leg and a robot trying to move across a foreign surface efficiently is the same problem: The robot has to adapt to a new environment. To do this, the Creadapt project is teaching robots to do what animals (like humans) do. Namely, we innovate, optimize, evolve, and adapt. Or, in other words, we try different things until we find what works, and then we keep trying variations on what works until we've found what works best.

The general philosophy here is to teach robots to be able to find their own solutions to complex problems. This is important for long-term reliable autonomy because there's no



way that we can possibly foresee all of the different problems that a robot might run into. If, for example, you have some robots that are exploring another planet and, oh, I don't know, one of them gets stuck in a sand dune or something, you'd want it to be able to try stuff until it manages to get free, learning from each attempt as it does so.

Creadapt has funding through 2016, so we're just starting to see what might be possible here. Eventually, algorithms that allow robots to adapt and recover could be everywhere, making them much more reliable than ever before.

spatial maps," he said.

While the UBC-Gavia has a relatively short range of 20 to 30 kilometers, the aim of the new AUV facility is to develop a fleet of robots fitted with an array of scientific sensors and capable of travelling much longer distances. It's an ambitious project and one that is not without risk.

"While working under ice is ideally suited for robotic exploration, it poses significant environmental challenges that require innovative engineering solutions," Dr. Forrest said.

"Every time we deploy we have unexpected challenges. However, what makes a good team is how they respond and overcome these challenges."

BAT BOT TO THE RESCUE

With a little bit of inspiration from the common vampire bat, Ludovic Daler has delivered an updated robot that not only keeps its walking ability, but now includes a foldable skeleton mechanism that lets it deploy and retract its wings.

The robot is called DALER (Deployable Air-Land Exploration Robot) and in the air, DALER can reach 20 m/s; on the ground, it tops out at 6 cm/s. It's not going to get very far at 6 cm/s, but in a search and rescue scenario, the ability to move on the ground at all means that it can land and scramble around a little bit, possibly moving underneath obstructions to investigate areas that are otherwise inaccessible from the air.



COOKING UP KNOWLEDGE FROM YOUTUBE

Imagine having a personal robot prepare your breakfast every morning. Now, imagine that this robot didn't need any help figuring out how to make the perfect omelet because it learned all the necessary steps by watching videos on YouTube. It might sound like science fiction, but a team at the University of Maryland has just made a significant breakthrough that will bring this scenario one step closer to reality.

Researchers at the University of Maryland Institute for Advanced Computer Studies (UMIACS) partnered with a scientist at the National Information Communications Technology Research Centre of Excellence in Australia (NICTA) to develop robotic systems that are able to teach themselves. Specifically, these robots are able to learn the intricate grasping and manipulation movements required for cooking by watching online cooking videos. The key breakthrough is that the robots can "think" for themselves, determining the best combination of observed motions that will allow them to efficiently accomplish a given task.

The researchers achieved this milestone by combining approaches from three distinct research areas: artificial intelligence, or the design of computers that can make their own decisions; computer vision, or the engineering of systems that can accurately identify shapes and movements; and natural language processing, or the development of robust systems that can understand spoken commands. Although the underlying work is complex, the team wanted the results to reflect something practical and relatable to people's daily lives.

"We chose cooking videos because everyone has done it and understands it," said Yiannis Aloimonos, UMD professor of computer science and director of the Computer Vision Lab — one of 16 labs and centers in UMIACS. "But cooking is



University of Maryland computer scientist Yiannis Aloimonos (center) is developing robotic systems able to visually recognize objects and generate new behavior based on those observations.

complex in terms of manipulation, the steps involved and the tools you use. If you want to cut a cucumber, for example, you need to grab the knife, move it into place, make the cut, and observe the results to make sure you did them properly."

One key challenge was devising a way for the robots to parse individual steps appropriately, while gathering information from videos that varied in quality and consistency. The robots needed to be able to recognize each distinct step, assign it to a "rule" that dictates a certain behavior, and then string together these behaviors in the proper order.

While robots have been used to carry out complicated tasks for decades (think automobile assembly lines), these must be carefully programmed and calibrated by human technicians. Self-learning robots could gather the necessary information by watching others, which is the same way humans learn. Aloimonos and Fermüller envision a future in which robots tend to the mundane chores of daily life while humans are free to pursue more stimulating tasks.

"By having flexible robots, we're contributing to the next phase of automation. This will be the next industrial revolution," said Aloimonos.



University of Maryland researcher Cornelia Fermüller (left) works with graduate student, Yezhou Yang (right) on computer vision systems able to accurately identify and replicate intricate hand movements.

Photos courtesy of John T. Consoli.





PRINTED ACTUATORS, ACTUALLY

How easy is it to build a robot? Not very. How easy could it be to build a robot? Well ... what if we try just sticking printable/inflatable actuators made of plastic onto origami (or anything else)?

As Ryuma Niiyama and colleagues from the University of Tokyo and MIT have discovered, it really is that easy.

The actuators get printed on a custom-made fabrication machine that's nothing more than a three-axis CNC holding a rod that can be heated (like a soldering iron).

Moving the iron over two sheets of thermoplastic bonds them, leaving patterns of interlocking inflatable pouches, and there's your actuator. Hooking up a tube to the actuator with a syringe on one end allows the user to inflate and deflate the actuator, which will drag along anything that you've decided to stick it to.

Anyone with a 3D printer could make these actuators in minutes to whatever specification they wanted.

PHONE FOR BRAINS

At CES 2015, Qualcomm presented their collaborative project with University of Pennsylvania researchers led by Vijay Kumar. It's a quadrotor that uses a smartphone for a brain for autonomous flight, using only onboard hardware and vision algorithms — no GPS. Impressive.

Continued on page 69

Recycling & Remarketing High Technology
WEIRDSTUFF®
WAREHOUSE
 Software, Computers, Electronics, Equipment, Doo-hickies
384 W. Caribbean Dr.
Sunnyvale, CA 94089
 Mon-Sat: 9:30-6:00 Sun: 11:00-5:00
 (408)743-5650 Store x324

WE BUY AND SELL EXCESS & OBSOLETE INVENTORIES!

FREE COMPUTER RECYCLING
 We recycle computers, monitors, and electronic equipment. M-Sat 9:30-4:00

GREAT DEALS!
 Hi-tech items, electronics test equipment, and more!

GIANT AS-IS SECTION
 10,000 sq. ft. of computers, electronics, software, doo-hickies, cables, and more!

also check out our...
eBay Store
stores.ebay.com/WeirdStuff-Inc
WWW.WEIRDSTUFF.COM

SERVO
SERVO BOOK SPECIALS

ARDUINO ROBOT SONANZA
 Gordon McComb
 413 Pages
Only \$23.95

SAVE 20%!

336 Pages
Only \$23.95
Androids
 Build Your Own Lifelike Robots
 Bryan Bergman, M.D., and Thomas B. Talbot, M.D.

To order call 800 783-4624 or visit:
<http://store.servomagazine.com>
 Limited time offer

ALL ELECTRONICS CORPORATION

Thousands of Electronic Parts Available Today!

LEDs • CONNECTORS • RELAYS
 SOLENOIDS • FANS • ENCLOSURES
 MOTORS • WHEELS • MAGNETS
 PC BOARDS • POWER SUPPLIES
 SWITCHES • LIGHTS • BATTERIES
 and many more items...

We have what you need for your next project.

www.allelectronics.com
 Discount Prices • Fast Shipping

SDP Stock Drive Products
 Setting Ideas Into Motion

One-Stop Shop for Mechatronic Components

Explore | Design | Buy Online

Over 100,000 small mechanical parts

www.sdp-si.com
 no minimum requirement

Designatronics inc.

COMBAT ZONE

Post comments on this section and find any associated files and/or downloads at www.servomagazine.com/index.php/magazine/article/march2015_CombatZone.

Featured This Month:

22 *A Veteran Builder Experiments with Brushless Drives*

by Kevin M. Berry with Hal Rucker

24 *CARTOON*

25 *BUILD REPORT: The Evolution of Slim Pickens*

by Nate Franklin

27 *PARTS IS PARTS: Product Review: AndyMark 7075 Aluminum Sprockets*

by Michael Jeffries

29 *Testing a Theory ...*

by Pete Smith

A Veteran Builder Experiments with Brushless Drives

● by Kevin M. Berry with Hal Rucker

Hal Rucker has lots of experience building combat bots using brushed motors. Over his 13 years in the sport, he's built two super heavyweights, two heavyweights, and two lightweights. Hal competes with his daughter at Combots, RoboGames, STEM Tech Olympiad, and a variety of other events where he certainly has had ample chance to test his designs in the box (<http://halrucker.com/robots>).

Recently, he spent some time exploring the use of brushless motors as drive motors. He knew that their power/weight ratio makes brushless the backbone of today's spinning weapons. They also come in enough sizes that one can find a good brushless weapon motor for any size bot.

After some tinkering and experimenting, he concluded he couldn't justify using brushless for his drive systems yet because:

1. Unsensored brushless motors

need complex electromechanical modifications to get them to spin forward/reverse and brake responsively.

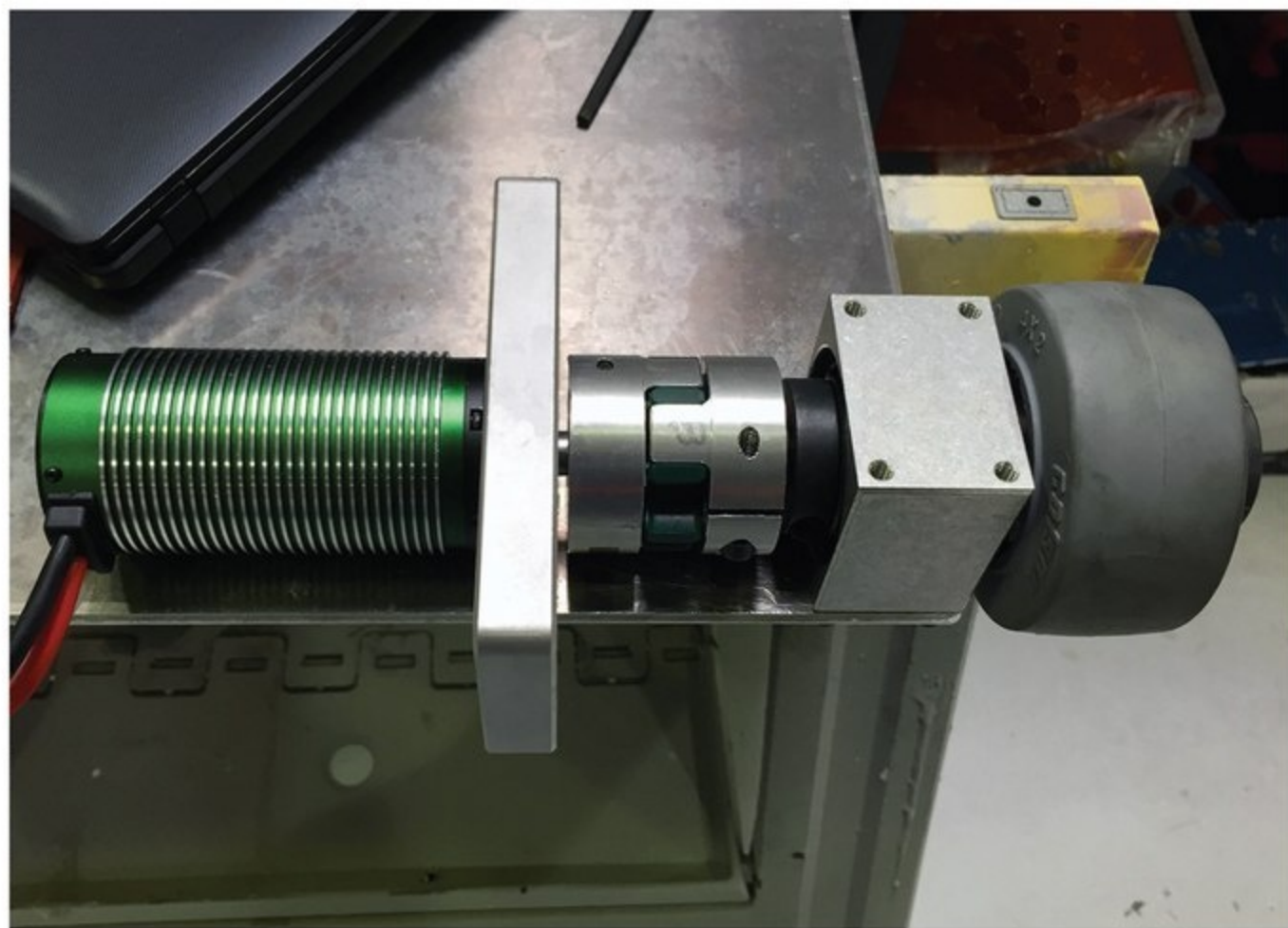
2. Sensored brushless motors are either too big, too small, too expensive, or too fragile for combat.

3. There are several battle-tested brushed motors that are powerful enough to spin a bot's wheels under a full load, so the additional horsepower one seeks in brushless motors is less exciting for driving than it is for spinning weapons.

Looking around the sport, he did acknowledge a couple exceptions:

1. Using brushless for drive on smaller bots that have big weapons. With the right gearing, a small brushless can move and point a big weapon around the arena (K2, Hot Stuff, Unmakerbot).

2. Running brushless EV motors in big boxy pushy bots with one motor



Full test rig including brushless motor, coupler, gearbox, and wheel.

per side. Hal could, for example, see two Motenergy brushless motors retrofitted into a bot like Vlad.

Although the large EV motors can be used well for drive, Hal couldn't find a low profile, powerful sensed system that would work for a heavyweight low profile design. Like a good gearhead, he didn't stop at research and reading spec sheets. He mocked up a gearbox, clutch, and special coupler to deal with the known challenges of brushless.

His findings? To quote Hal, "The modifications all sorta helped and sorta worked ... but their added complexity introduced the kind of gadgetry that performs perfectly during testing and fails catastrophically during a fight."

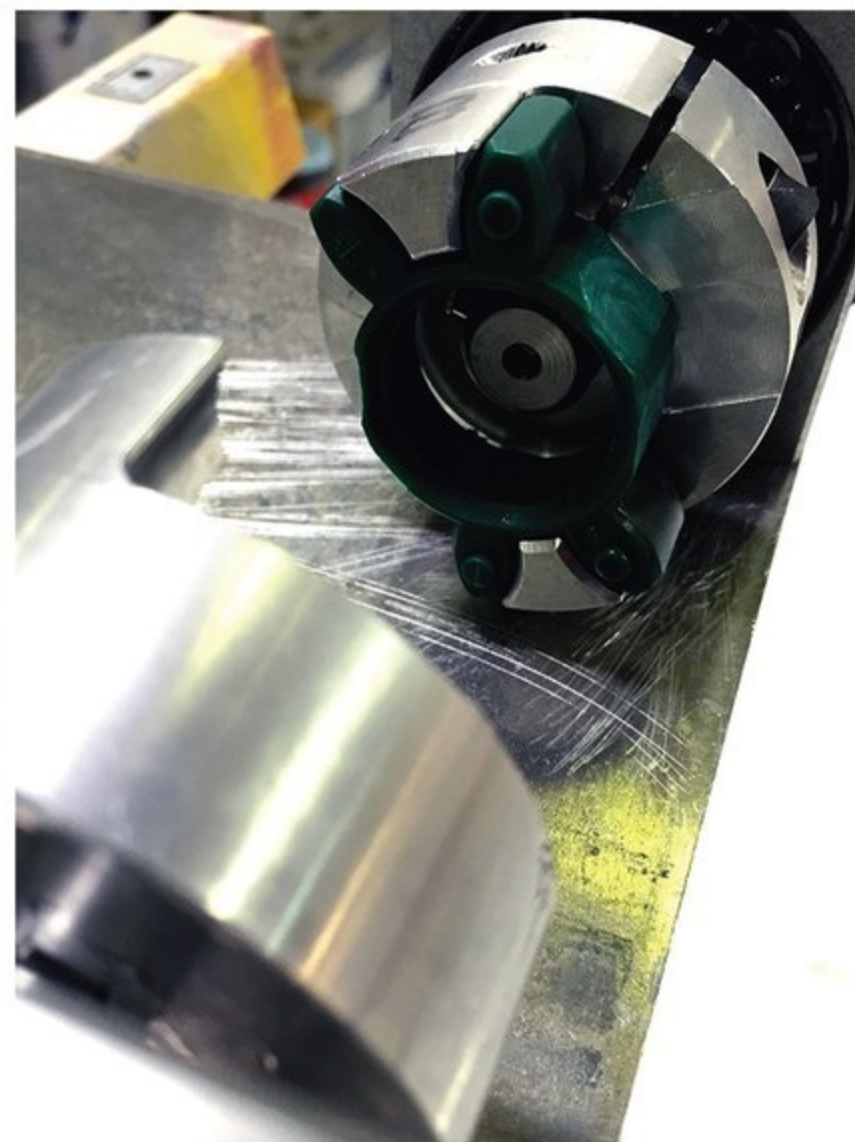
Anyone who's taken an elegant design into combat knows that brute force often triumphs over fine design. In the interest of increasing the community's knowledge — in that weird way we help each other further the arms race — he agreed to share his experiments and findings with the Combat Zone's readers.

Typical brushless motors have very

little torque when they start to spin. If there is initially a large load on the motor, the motor has a strong cogging torque, or the sensorless system isn't sure about how to spin at the moment, the unit often doesn't have enough torque to start spinning. Or, the motor will rotate back and forth in a jerky random pattern. To overcome this problem, Hal first hacked together a motor coupler that rotated freely about 100 degrees before engaging the drive shaft. (He started with a spider coupler and removed two of the forks and two of the rubber pads.)

Next, Hal built a gearbox using low pitch spur gears. The play in the gears also helped with the startup problems. The final test setup was a Castle Creations Mamba XL2 ESC and 2028 brushless motor coupled to the gearbox input shaft with the modified spider coupler.

Most of today's quality hobby brushless controllers include software or a programming card that enables the builder to easily set key parameters including forward, reverse, and braking. This makes it easier to set up a brushless motor to behave more like a brushed motor that likes



Partially assembled coupler, showing how 100 degrees of rotation is allowed.

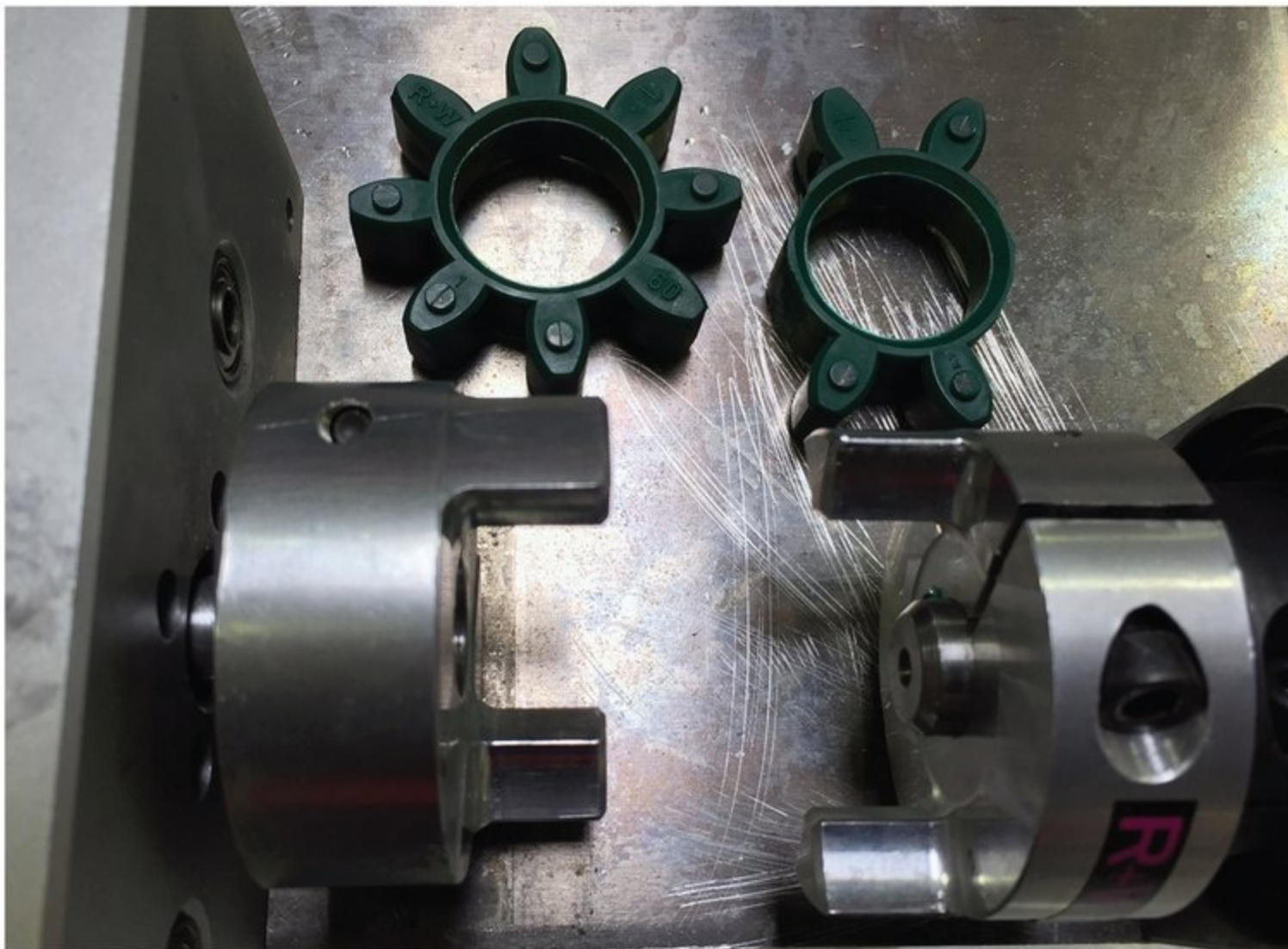
to spin in both directions and quickly brake on a dime. The more powerful controllers like the Mamba XL2 just don't seem happy, though. The 2028 motor would rather spin very fast in one direction.

In Hal's setup, he was initially able to get full forward/reverse/brake operating properly, but the angry sounds emitted by the motor were worrisome.

He added: "Once I put any real load on the output of the gearbox, cogging became an unpredictable problem again. I could confuse the controller just by gripping the output shaft of the gearbox with my gloved hand."

Later, Hal tried a centripetal clutch idea that engaged the motor after it achieved a certain speed. He doesn't have photos but, as he put it, "I knew once I got into the build I was looking at a lot of time and money to refine a custom clutch. My old brushed motors were looking just fine at this point."

In conclusion, Hal strongly



Custom coupler, fabricated from a spider coupler with forks and pads removed.

Some useful links Hal recommended:

Maxon Motors

www.maxonmotorusa.com/maxon/view/content/products

Castle Creations Motors and ESCs

www.castlecreations.com/products/products_drive.html

NeuMotors Custom Brushless Motors

www.castlecreations.com/products/neumotors/nm.html

MGM Brushless Controllers
(hobby and professional)

www.mgm-controllers.com

Roboteq Brushless Controllers

www.roboteq.com/index.php/roboteq-products-and-services/brushless-dc-motor-controllers

believes a smart builder is going to put together a HW brushless drive system soon that will have lots of reliable power, but he's decided not to pursue it any further until a vendor offers a solution he can buy off the shelf.

"In my mind, this would be a 5+ horsepower sensorless motor, three

inch or less diameter with a 36V 300A brushless ESC that knows what to do with forward/reverse/brake and won't whine about it. A sturdy motor can — with magnets that won't pop off and at least a 1/2 inch output shaft."

We know that many suppliers of combat parts read the Combat Zone,

and hope some are paying attention! Then again, most of the hardened solutions to our unique problems come from the builders. After testing in the box, they tend to find their way into everyone's hands, so maybe in a year or two, we'll all be running brushless in the box. **SV**

Melty Brains

March Madness

by Kevin Berry

Who do you think will win at WAR? Allosouras or Angry Radish?

Huh? Are they Big 12 or SEC?

Why Botheads don't go to sports bars ...

EVENT REPORT: The Evolution of Slim Pickens

● by Nate Franklin

In the February issue of *SERVO Magazine*, I talked about my first event and the creation of my first robot, Slim Pickens. This article will talk about how Slim Pickens has changed over the years, and what I've learned along the way.

Following my second place finish at Franklin 2012, I set my sights on the next event: Motorama. From the videos I saw at the time, I knew that the competition would be tough. At that point, I still had no idea where to go next, so I decided to continue doing what I knew how to do. I replaced the Viper kit base on Slim Pickens and bought some more sheet metal from the hardware store. The new armor was slightly thicker, but it was literally the same bot (**Figure 1**).

At Motorama, Slim Pickens went 1-2, losing to two wedges in two close matches. I threw my bot into the rumble, and it didn't last long. Horizontal spinner, Vile Ant (**Figure 2**) got behind me and ripped out my power switch right in the first few seconds. Slim Pickens just lay there in the arena taking hits from everyone in



Figure 1.

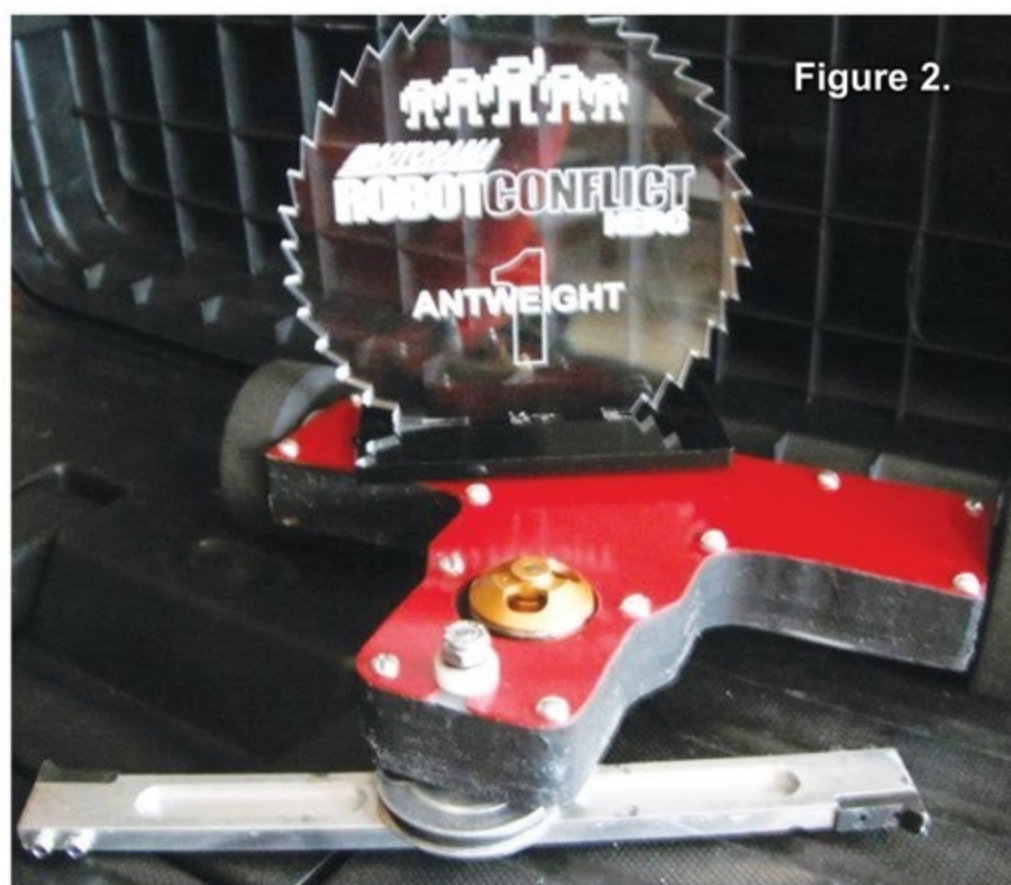


Figure 2.

the rumble.

One of the main problems with Slim Pickens at this point was the batteries I was using. It ran on two nine volt Energizer batteries as instructed in the kit's manual. This meant that I barely got enough juice as compared to the other bots.

The next event came — Bot Blast 2013 — and Slim Pickens was slightly different. Aside from having a new battery, it had an extra plate on the wedge that I thought would help keep it slightly more protected (**Figure 3**).

This event turned out to be my worst.

Slim Pickens lost both of its matches, and I threw it in the rumble. For the rumble, I added little horns to try and keep the other bots from getting on top of me while I was pushing them (**Figure 4**). After the five minutes were over, Slim Pickens was chewed up but still mobile. After Bot Blast, I learned that thin sheet metal wasn't going to cut it against the powerful spinners (**Figure 5**).

Over the next few months, I was talking to fellow bot builder, Alex Horne

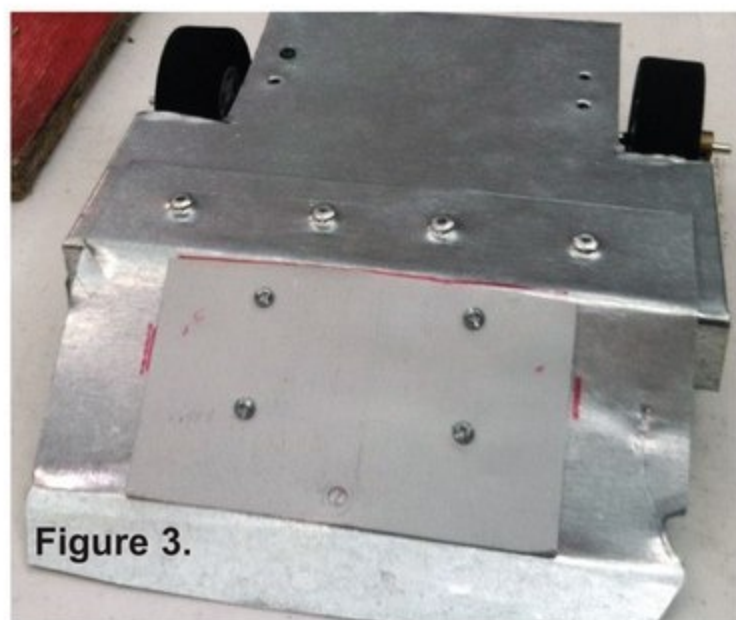


Figure 3.

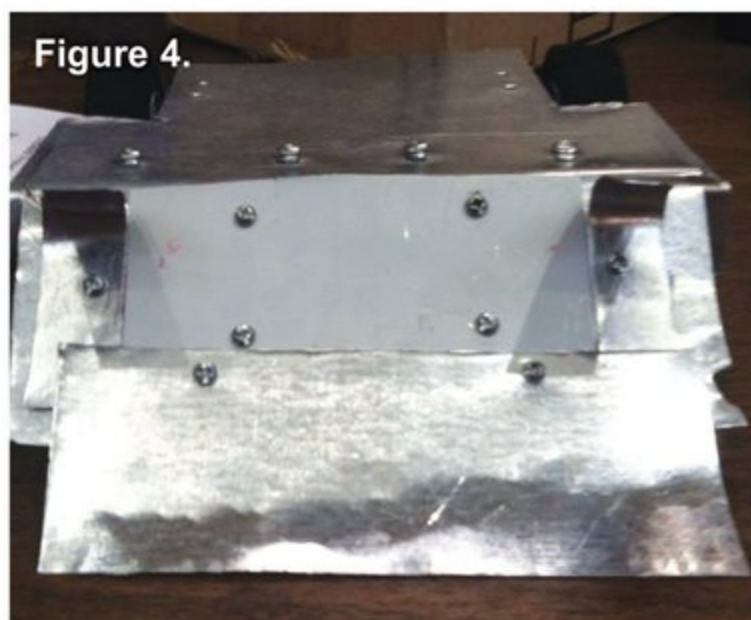


Figure 4.

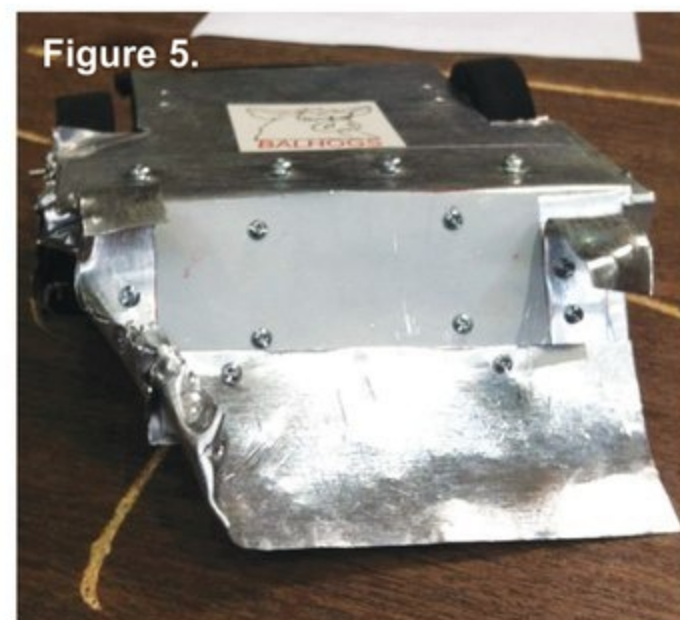


Figure 5.

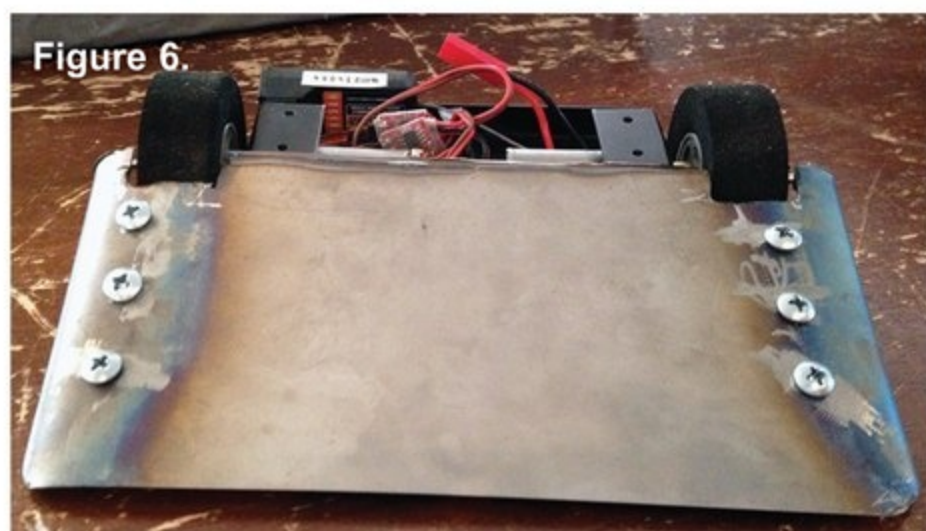


Figure 6.

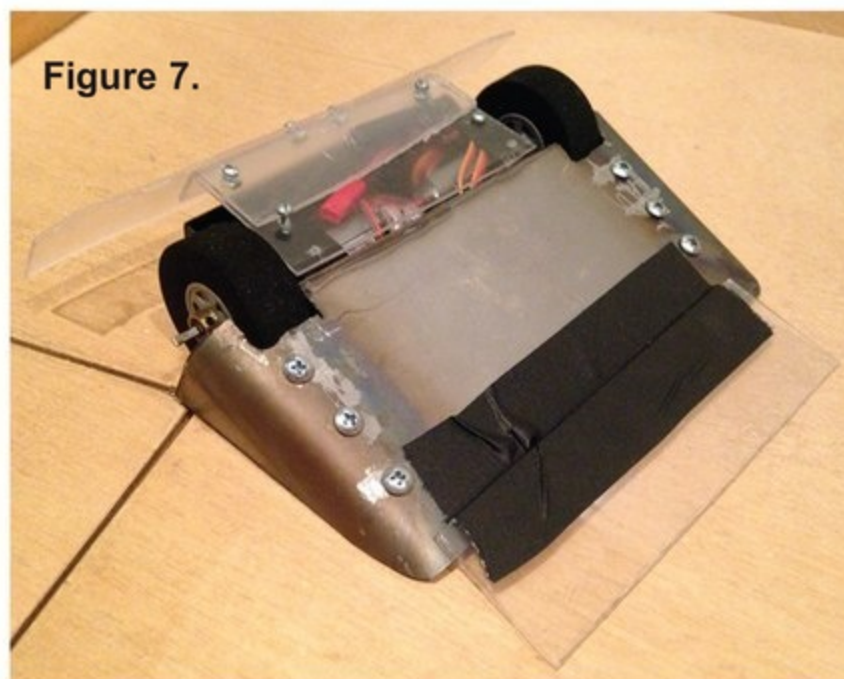


Figure 7.



Figure 8.

about what to do. He said he'd help me out with rebuilding Slim Pickens. The same Viper base kit was used, but was modified to save weight for a new .04 inch titanium wedge (**Figure 6**).

The wedge wasn't perfect. It was warped to the point in which it wouldn't touch the ground. I had to give it a hinged wedge made out of lexan and gaff tape. The bot's back cover was sloppily made by me since I was pressed for time and wasn't able to make a titanium one (**Figure 7**).

Another change for Slim Pickens was the gear ratio for its motors.

getting underneath the other bots.

Another issue with the wedge was that it got stuck under the arena bumper (thankfully, the other bot got stuck too, so both of us were unstuck). After this event, it was back to the drawing board once again.

The Viper element of Slim Pickens was abandoned. The base was too thin and I didn't want to keep having to purchase them after each event. The new Slim Pickens was inspired by Kurtis Wanner's Antweight, Kitbot, which had received the bronze medal at RoboGames 2013 (**Figure 8**).

Instead of the Viper's default 50:1 Silver Sparks, I was running 22.2:1 motors.

This version of Slim Pickens was quickly abandoned after going 1-2 at Franklin 2013. The main problems of the bot were that the hinged wedge would get flipped upward and would prevent me from

the event (Motorama 2014), all of the electronics burned out for some unknown reason. The other builders were able to give me their spare speed controllers and batteries just minutes before safety checks started. Slim Pickens was ready to go. This last minute miracle was able to get third place in the event after facing some tough competition.

For the next event — Bot Blast 2014 — Slim Pickens would see few upgrades. I gave it a new power switch and a shiny pair of FingerTech snap hubs (**Figure 10**). It managed to go 4-1 for second place. When October came by, Slim Pickens returned to Franklin to go undefeated for the first time.

Slim Pickens was my first bot, and is currently my most successful bot. It was also the bot that taught me the most. I learned what things worked and what didn't. I learned that no bot is perfect and will always be going through some sort of evolution — even when you think you've got it down.

Most importantly, I learned not to give up when you think you've totally flopped and can't recover.

Not only did Slim Pickens change, but I changed on my crazy journey throughout this sport, from all the highs and lows along the way. **SV**

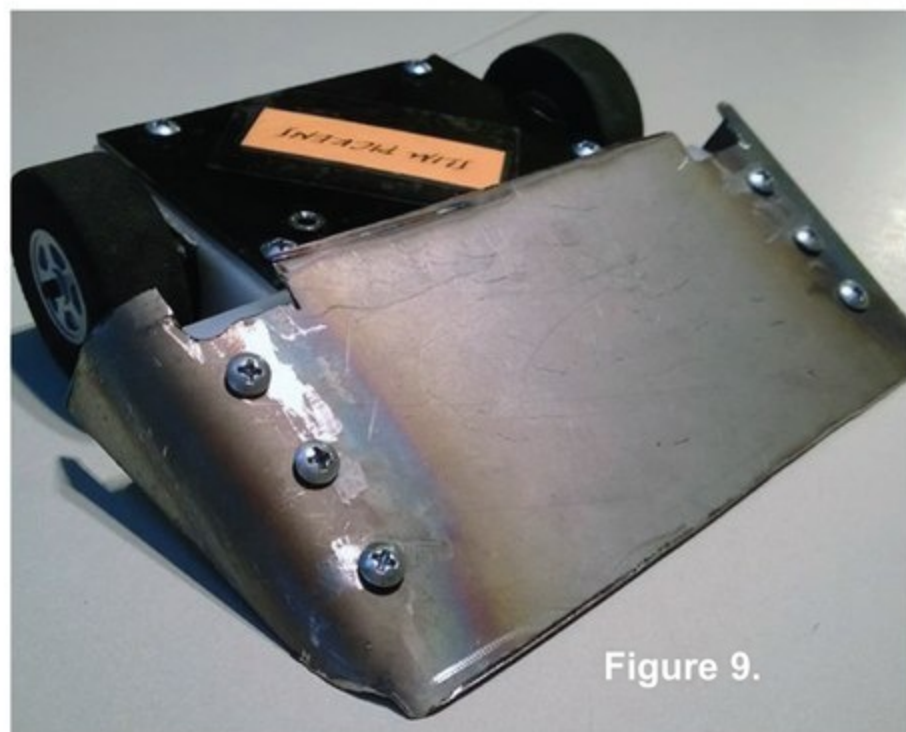


Figure 9.



Figure 10.

PARTS IS PARTS:

Product Review: AndyMark 7075 Aluminum Sprockets

● by Michael Jeffries

AndyMark (www.andymark.com) primarily focuses on providing components for the US FIRST Robotics Competition. However, in creating and supplying components for non-combat 120 lb robots, they have ended up creating a wide range of products that are suitable for use in combat bots across a wide range of weight classes. These components include wheels, gearboxes, motors, gears, hubs, and sprockets. In addition to this, they're also the main distributor for the newest generation of the popular BaneBots planetary gearboxes.

For the last few years, I've been using one of their simpler products with a great deal of success. AndyMark sells #25 and #35 roller chain sprockets made out of 7075 aluminum. The sprockets are lightweight, compact, and reasonably priced (ranging between \$9 and \$16, depending on sizing).

My first time using the 7075 sprockets was on the initial build of my 30 lb Sportsman class robot, Nyx. Initially, I used the 15 tooth 7075 sprockets to drive the wheels. The sprockets themselves worked flawlessly. However, after a few events, I switched to 12 tooth AndyMark sprockets to get a bit of extra acceleration and pushing power. As with the 15 tooth sprockets, the 12 tooth sprockets worked without issue for the remaining life of the robot.

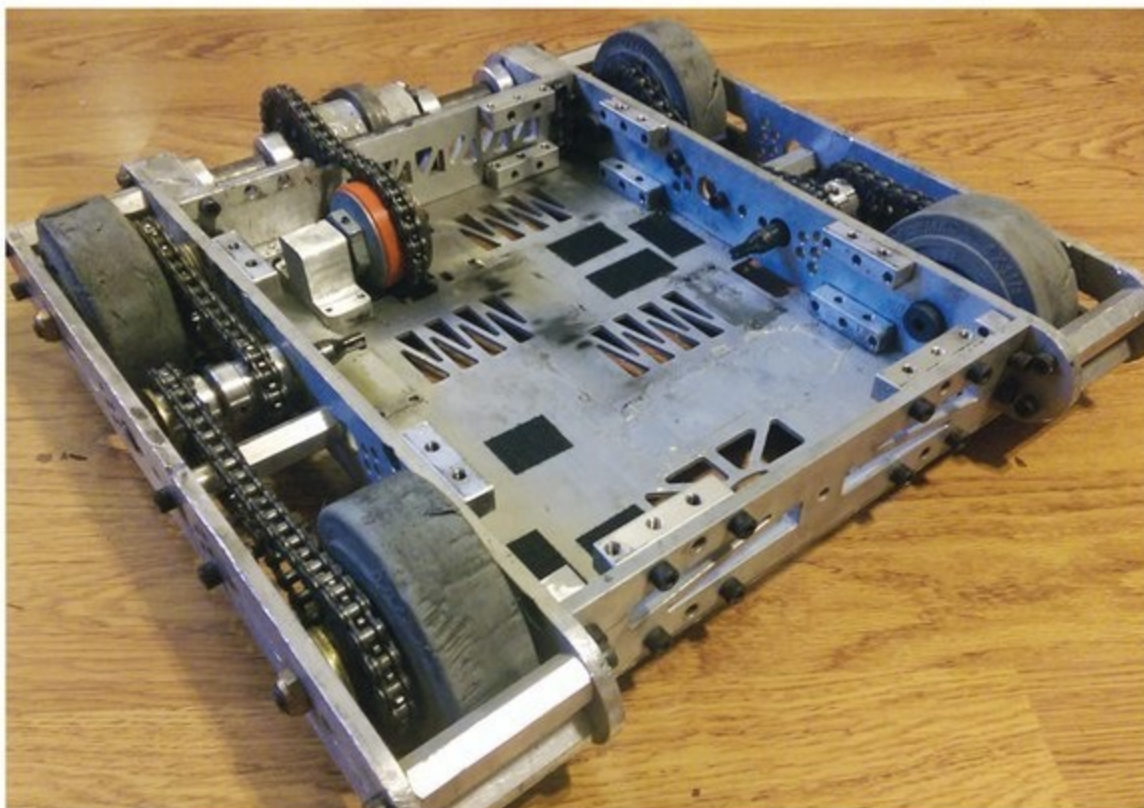
When it came time to build Spanky, I knew I wanted to gear down the drive motors a bit because even in low gear the robot would have been too fast with 8" wheels. Again, I used the AndyMark 7075 aluminum sprockets. In this case, I put the 12 tooth sprocket on the drive motor shaft and the 15 tooth sprocket on the wheel shaft. This provided an additional 1.25:1 reduction and made



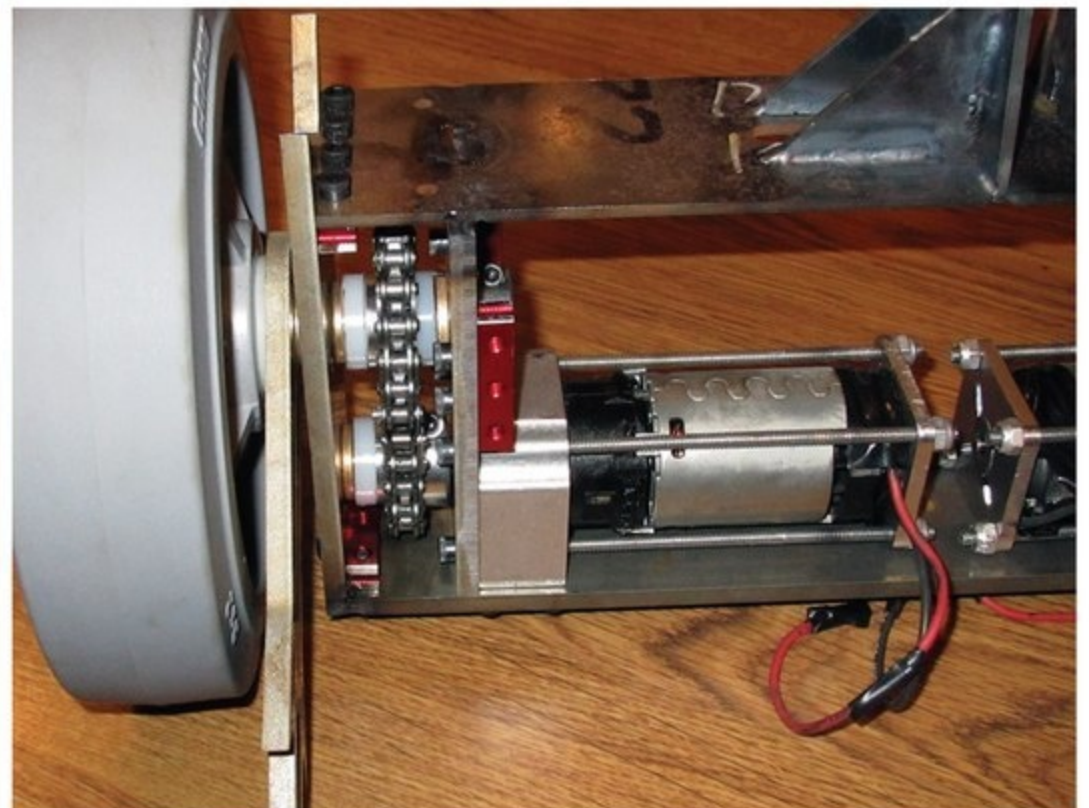
A 15 tooth #35 sprocket with a 1/2" bore and keyway made from 7075 aluminum.

the robot a bit more controllable. The compact profile of the sprockets also allowed me to design the chain reduction area to be quite compact, which saved weight for other aspects of the design.

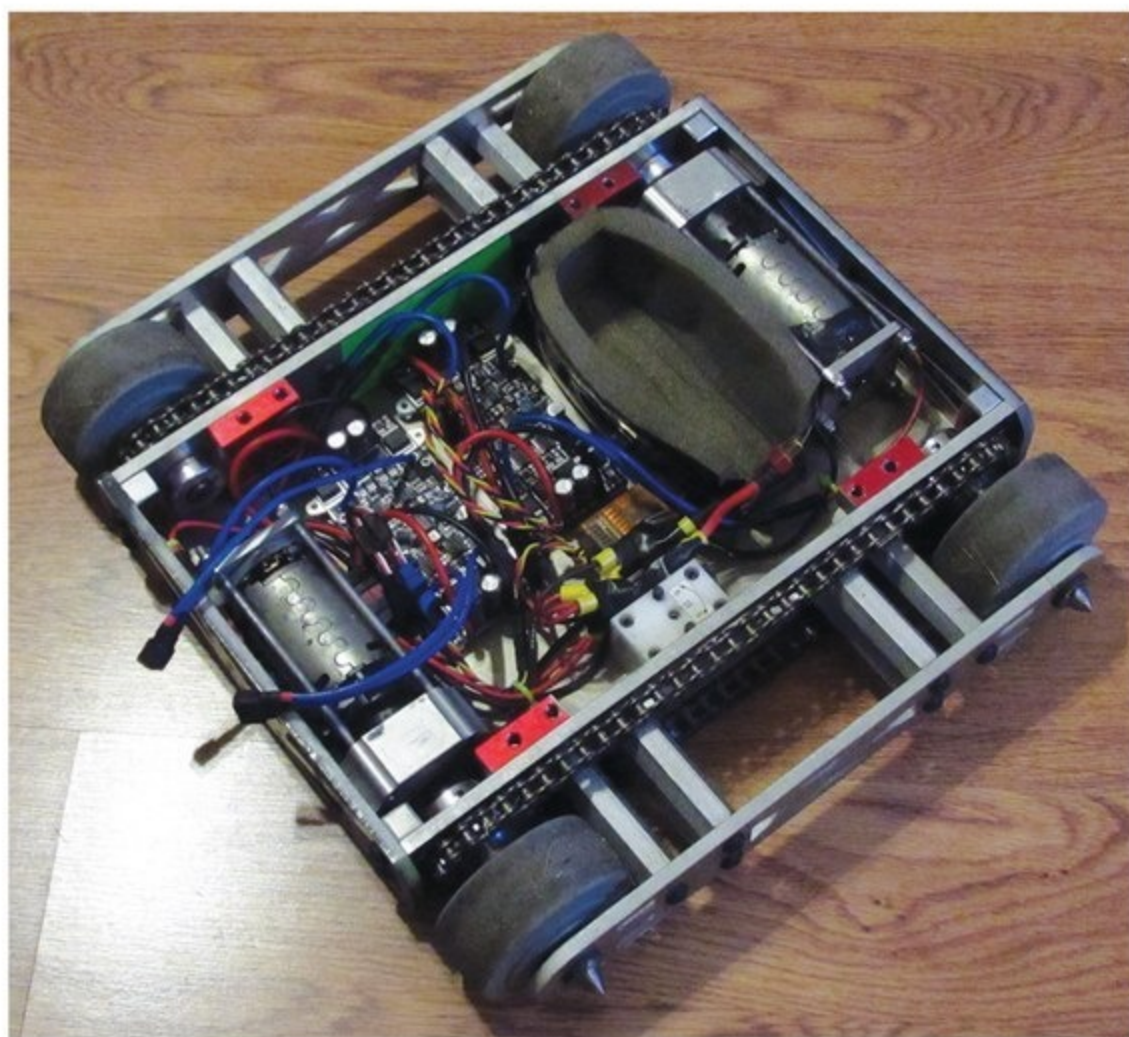
With how well they performed in



The first Nyx used 12 tooth 7075 sprockets to drive the Battlekits 4" wheels.



Spanky uses 12 and 15 tooth 7075 sprockets to add a bit of extra gear reduction between the drive motor and 8" wheels.



The new version of Nyx uses 15 tooth sprockets to give it four-wheel drive while using an offset motor configuration.

the original Nyx, I knew I would be using them again for the new version. In this case, I'm no longer using the sprockets to provide additional gear reduction. They're now being used to provide for a four-wheel drive platform while only using two drive motors. With how little wear they had shown, I was able to re-use the original 15 tooth sprockets I had purchased for the first version of Nyx.

The new Nyx is going to be competing at its first event in the near future; throughout extensive testing, the sprockets have again proven to be highly durable and trouble-free.

Overall, these sprockets are a great value and provide a compact lightweight sprocket solution that isn't available elsewhere. They're well worth the cost as they've proven repeatedly that when they're used properly, they have a life span measured in years.

The two specific sprockets I have tested are:

www.andymark.com/product-p/am-0166.htm

www.andymark.com/product-p/am-0167.htm

AndyMarks' other 7075 aluminum sprockets appear to be of the same quality and will likely perform just as well should you need different sizes. **SV**

The Robot MarketPlace

Supporting Robotic Combat for over 10 years!

- Combat Robot Supplies
- Motors
- Batteries & Chargers
- Vex Robotics
- Carbon Fiber
- Wheels & Tires
- Speed Controllers
- R/C Systems
- Drive Components
- Hobby Supplies & Toys

(877) ROBOT 99
(877-762-6899)
or (941) 749-6030

Your One-Stop Robot Shop!

RobotMarketPlace.com

WORLD'S MOST VERSATILE

CIRCUIT BOARD HOLDERS

Model 324 Our Circuit Board Holders add versatility & precision to your DIY electronics project. Solder, assemble & organize with ease.

VISIT US ON  

MONTHLY CONTEST
Visit us on Facebook® to post a photo of your creative PanaVise project for a chance to win a PanaVise prize package.

PANAVISE®
Innovative Holding Solutions

Model 201

7540 Colbert Drive • Reno • Nevada 89511 | (800) 759-7535 | www.PanaVise.com

Testing a Theory ...

● by Pete Smith

It was just after Motorama 2014 that I bought a weapon disc that had been designed for use on the lethal 30 lb vertical spinner, Shaka. Made from hardened S7 steel and weighing 8 lb (**Figure 1**), it was at a price I could not resist.

I decided that I would build a 30 lb horizontal spinner around this weapon. Why horizontal? Well, with the vertical spinners, Shaka and Hyper Active already in this weight class, I wanted something a little different. Horizontal spinners are also less affected by gyroscopic effects, so are easier to drive. Plus, with a much wider "kill zone," the opponent has a lot more problems avoiding the weapon.

The disc at 8 lb is not particularly heavy for a 30 pound bot, so to compensate for that lack of heft I wanted a fairly high RPM. My Hobbyweight (12 lb) horizontal blade spinner spun at about 4,000 RPM, but that was with a 14" blade. So, with the new disc only 11" in diameter, I decided to aim for 8,000 RPM.

I also wanted to keep the blade low to the ground to try to get under opponent's armor and get at their wheels more easily. To do this, I would use a live axle with a rounded base where it contacted the ground so the blade could be mounted below the bot. The spinning axle gets a little grip on the floor of the arena, so it only has a minor effect on controlling the bot.

The bot chassis would also have to be as thin as possible so that it could run inverted even with wheels as small as 4" in diameter.

It was this last requirement that prompted the test chassis that this article will describe. Traditionally, a combat bot has only one drive motor for the weapon, and the dimensions

of that motor will drive the final dimensions of the bot. Bigger brushless motors have a low kV (the RPM per volt), so to get the required weapon RPM I would have to use a much higher voltage than the commonly used 6S LiPo 22.2V, which then complicates the drive system as drive ESCs (electronic speed controllers) are not readily available for more than 6S. There are ways around this, but they add complications.

I then had the idea of using multiple smaller motors instead of one bigger one. They have a higher kV, so getting the required RPM is easier to achieve with 6s. Plus, being smaller, fitting them in should be easier.

I decided to try using three NTM 42-38 750 kV motors from HobbyKing. They would spin at 16,650 RPM (750 x 22.2) at 6S, so a 2:1 speed reduction in the weapon drive would produce the 8,000+ RPM desired. Each motor would be powered using a Turnigy Plush 60A ESC (I had used the 80A versions with success in the past).

I would need a chassis to mount the three motors and the blade itself to. Luckily, I was training at the time to use the Tormach CNC milling machine at my workplace. I used the opportunity to "CNC" machine a recess into a bar of 6061 aluminum, and added two fixed sets of mounting holes and one set of slots so that one of the motors could be moved to tension the timing belt that would transmit the power to the blade.

The blade would be attached to a 3/4" shoulder bolt mounted in matching needle roller bearings. These bearings were pressed into a reamed



Figure 1. Weapon disc from Shaka.

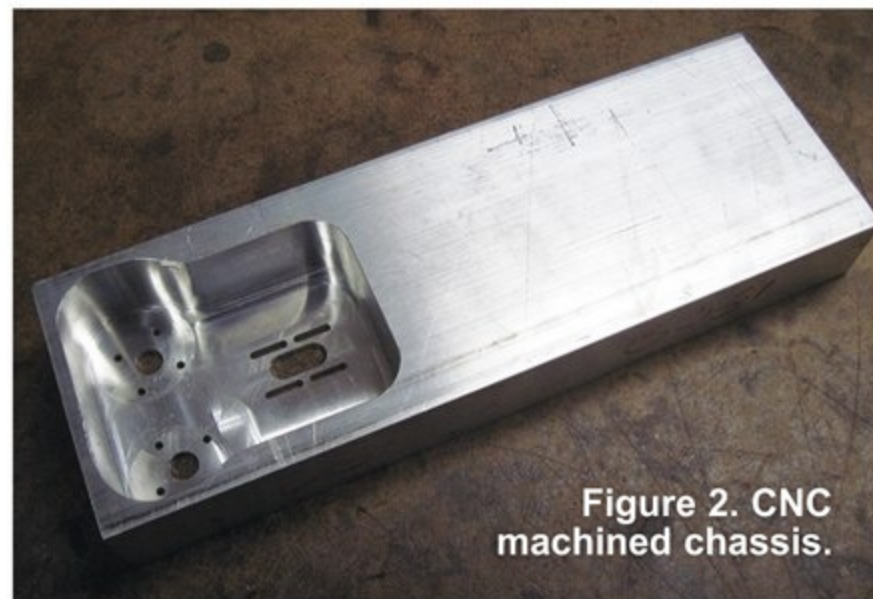


Figure 2. CNC machined chassis.



Figure 3. Main shaft roller bearings.

1" diameter bore at the other end of the chassis block (**Figure 3**).

The shoulder bolt has a finely ground surface that works well with roller bearings and the required keyways (**Figure 4**) to lock the blade. The drive pulley to it can be added using a carbide end mill the same width as the key — 3/16" in this case



Figure 4. Keyways in shaft.



Figure 5. Rounded head of shaft.



Figure 6. Loctite set screw.

— being careful not to have the keyways running into the surface used by the roller bearings. The end of the shaft was rounded off (**Figure 5**) using my lathe and a 4-1/2" angle grinder.

I prepared the motors by Loctiting the set screw in the end bell (**Figure 6**). Then, after wrapping the motor in tape (to keep metal filings out), I ground flats in the shafts (**Figure 7**) with a Dremel type tool.

The ESCs usually need to be programmed before use in combat robots. You can program them without the appropriate card, but since this involves listening for and counting various numbers of beeps and squeaks, I've found it much better to get the programming card that matches the motors. This makes programming easy (**Figure 8**).

The motors were then mounted

in the chassis using hex head screws and washers (**Figure 9**). It's important to carefully check that the screws are not too long as they can easily damage the windings in the motor. The chassis was assembled together, with the blade disc mounted on the axle with thrust roller bearings (**Figure 10**) and matching hardened washers between the blade and the chassis, and also between the chassis and the main weapon drive pulley (a how-to article about the pulleys is in the January 2015 issue of *SERVO*).

There were not enough threads left exposed on the end of the shoulder screw to allow the big Nylock

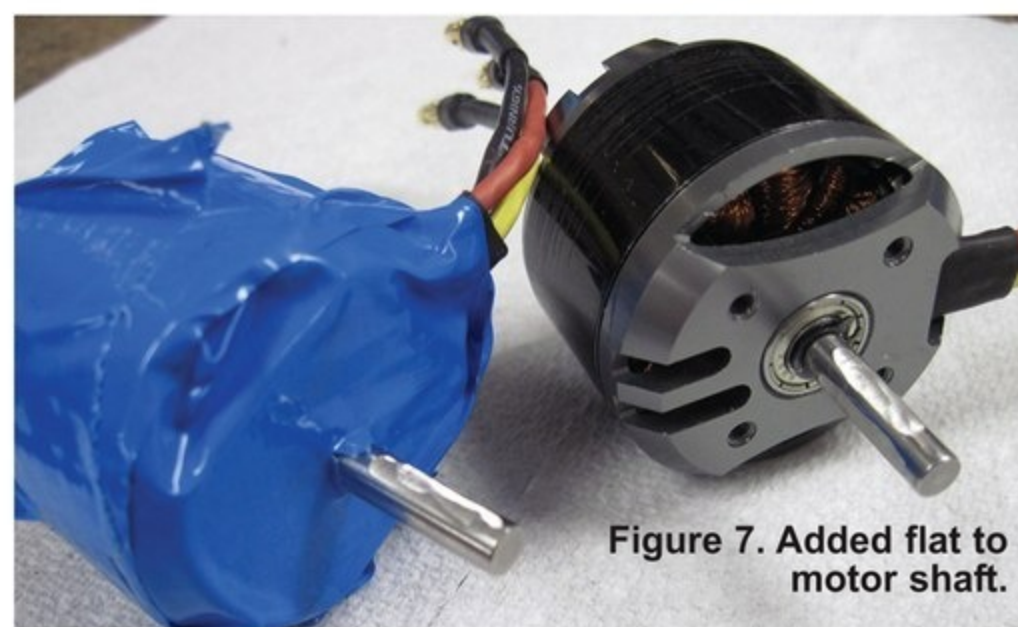


Figure 7. Added flat to motor shaft.

locking nut to fully engage, so I used Loctite on those threads to ensure it would not loosen up during testing. The nut was tightened just enough to allow play in the shaft to let it turn freely. I added some thin machine oil to the main and thrust bearings.

The completed test chassis (**Figure 11**) uses a double-sided 5 mm HTD belt to wind between the motors

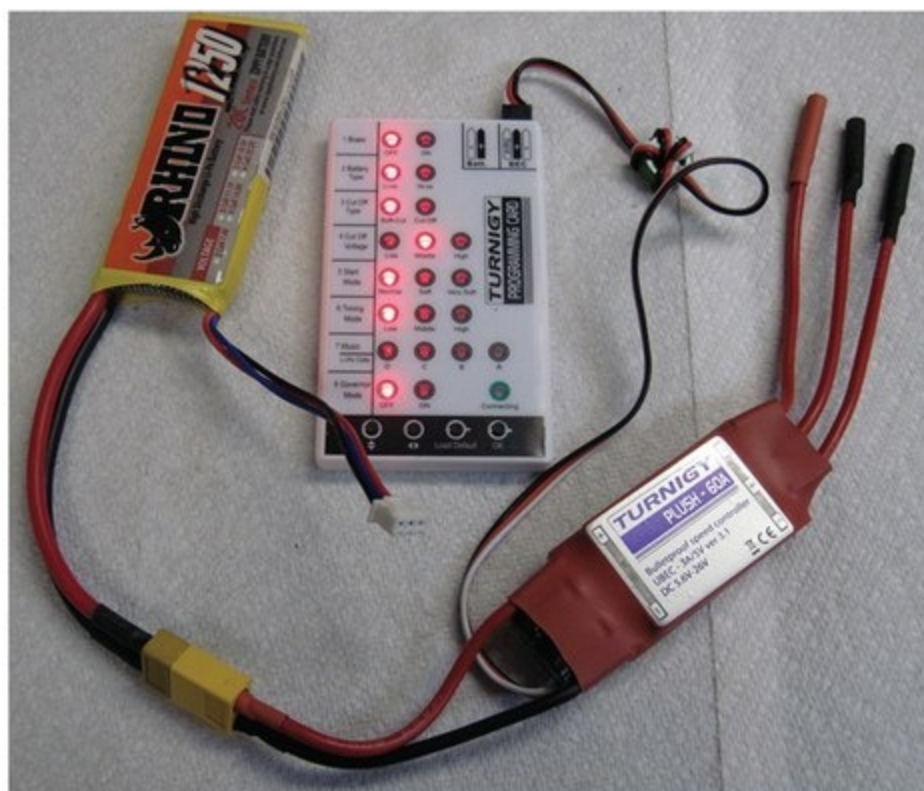


Figure 8. Programming ESCs.

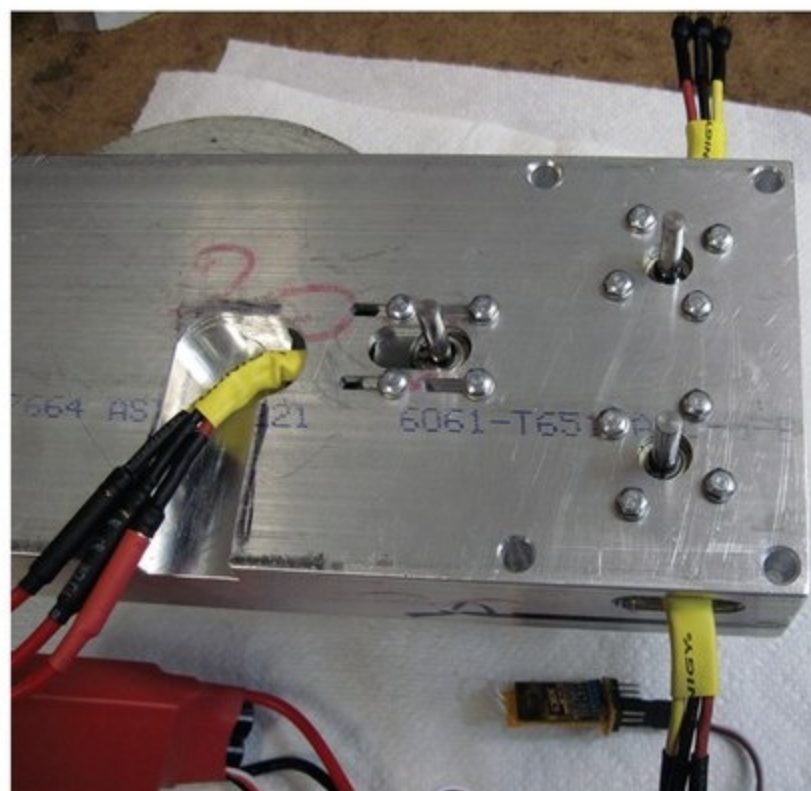


Figure 9. Motors mounted.

and then up to the weapon pulley. The middle motor can move in the slots to tension the belt. The hex mounting screws on the motor allow the screw to be tightened with a small wrench as the pulley would block access to a Phillips or socket head.

Tensioning the belt proved to be trickier than expected, and getting the right balance between

having the belt tight enough to not jump a tooth and loose enough to allow easy rotation was not easy. If I was building the test chassis again, I would mount all three motors on one plate and have the whole plate move to tension the belts. The plate could be secured with much larger and more accessible screws this way. A

rectangular tube was bolted across the back of the main chassis bar to allow it to be screwed down to a base.

To make testing a little safer, I built a small test chamber out of a heavily reinforced plywood box that had once contained a rotary table for my mill. A greased metal plate was screwed down under where the axle would rub. The 1/2" plywood walls and 3/8" thick polycarbonate top panel should at least slow down any projectiles somewhat.

The motors were hooked up with their ESCs (the middle motor must rotate in the opposite direction from the other two), and all three ESCs were connected by a custom three-way signal lead to the Rx. Tests at slow speed proved successful, however, even at one third power, it was scary!

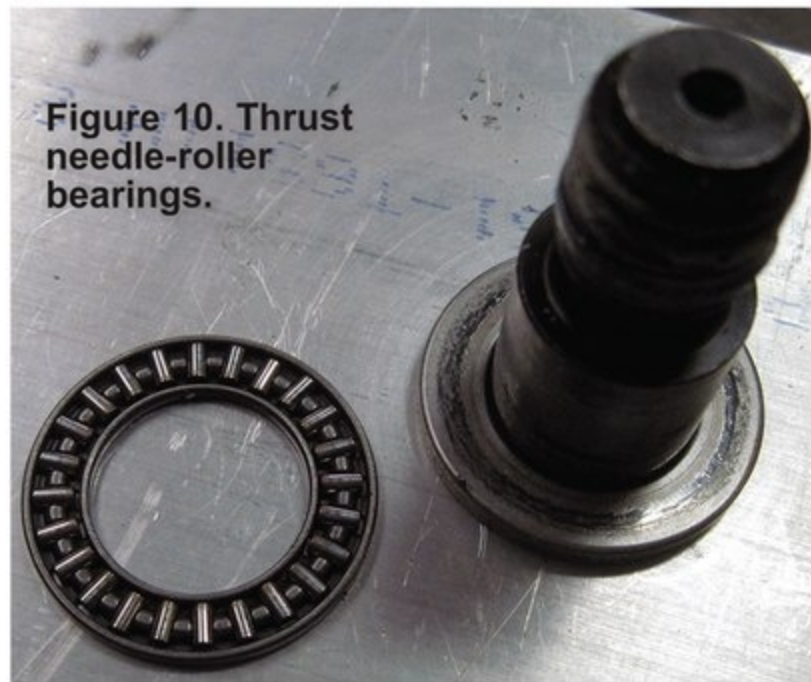


Figure 10. Thrust needle-roller bearings.

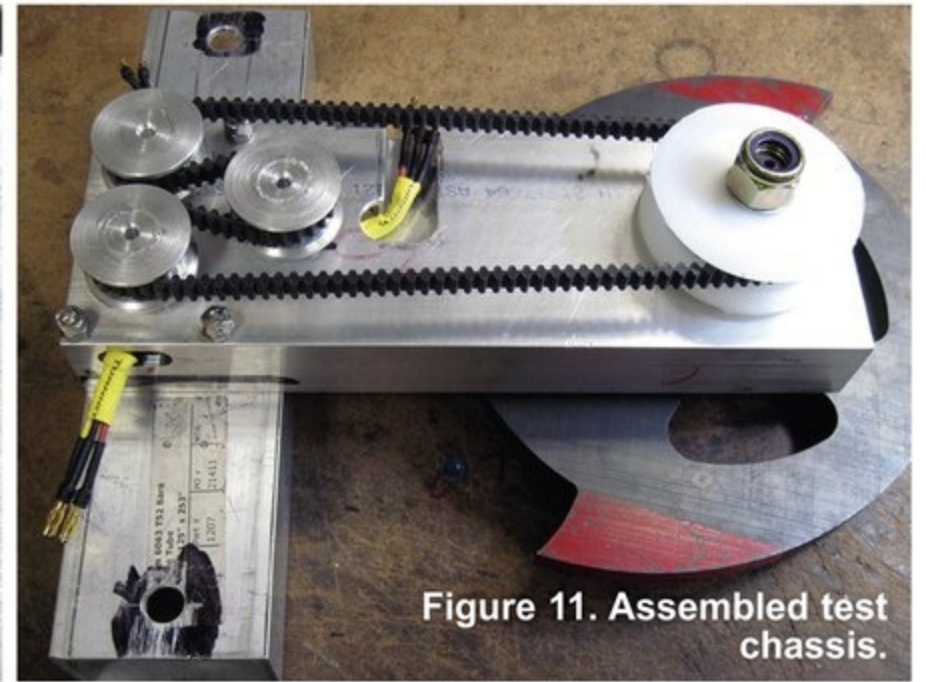


Figure 11. Assembled test chassis.

Moving it out to our back garden in the woods, I brought it up to one half power from a safe distance. I could see through the clear top panel that the belt seemed to be "surging." What I should have done was stop it and re-tension it (it may have stretched somewhat or maybe even just became more flexible with a little use), however I pressed on.

At about two thirds power, there was a small bang. The belt came off and was torn into two pieces, plus, one of the motor shafts had failed (**Figure 13**).

I'm not sure what happened first — did the motor shaft fail and then the belt got caught in the disc? Or, did the belt breaking first result in the motor shaft break?

Although the test was a failure, it was much better for the design to fail in a cheap test chassis than in a very

expensive completed bot. The three motors certainly had no problems quickly powering up the disc, showing that it had enough power, but I had not considered how that total power may have been too much for the relatively small diameter shafts on the motors. Or, what about when the bot gets a big hit on an opponent? Would an even better quality replacement shaft without a circlip groove (where the shafts usually fail, and did so this time) be able to survive the sudden deceleration?

I will have to rethink the weapon drive for my new bot. The motors and ESCs from the test will likely end up in a 12 lb Hobbyweight, so they won't be wasted. It looks like I may have to go to a single motor at a higher voltage on my new 30 lber.

Oh well, back to the drawing board ... **SV**



Figure 12. Test box.

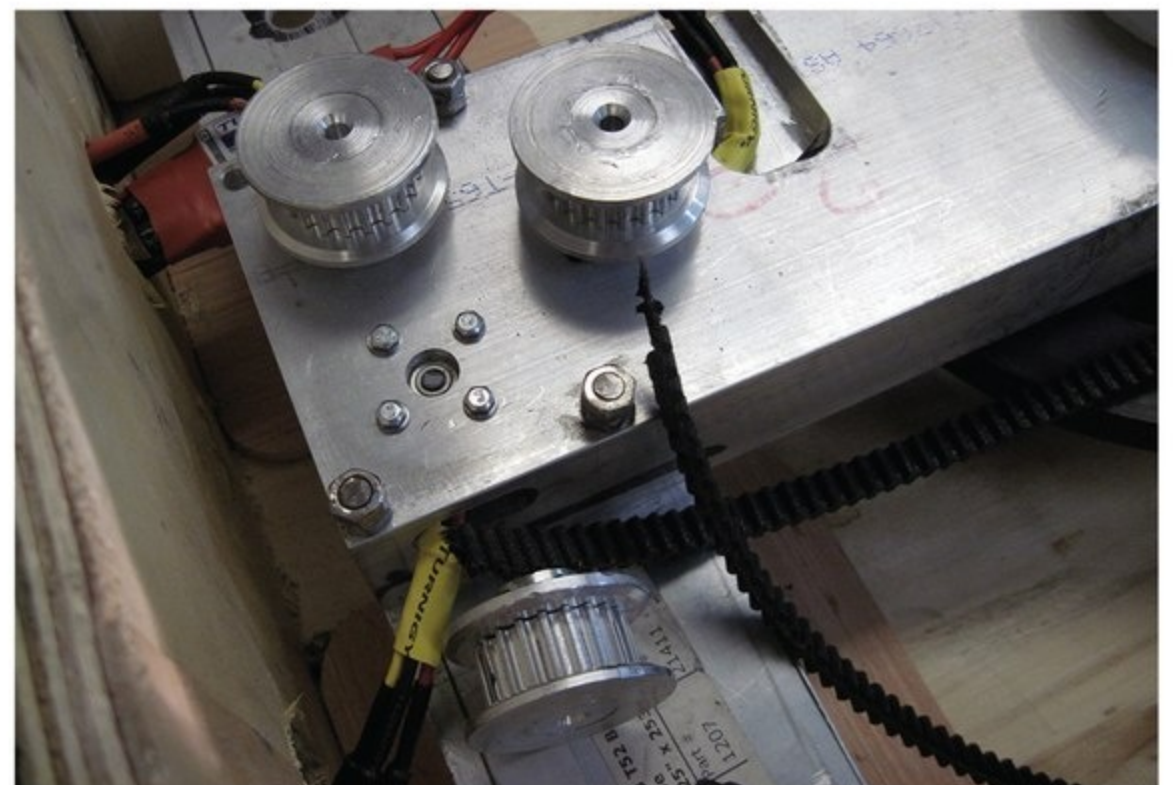


Figure 13. Belt and pulley after test.

The Robot You've Always Wanted

Part 3: Programming Navigational Behaviors

By John Blankenship and Samuel Mishal

The life-sized Arlo robot discussed in Parts 1 and 2 of this series has 27 real and nine virtual sensors — far more than most hobby or educational robots. This article discusses how this robust sensory system can be used to handle complex situations such as navigating throughout your home. Although Arlo is controlled by RobotBASIC, a navigating robot could be controlled by a Propeller, a Raspberry Pi, or nearly any other processor implementing the algorithms discussed.



Figure 1.

One of the reasons Arlo (see **Figure 1**) is the robot you've always wanted is that he can do things that actually impress your friends — even friends that don't personally play with robots. One such behavior is being able to autonomously navigate from room to room in your home. Let's explore some simple — yet effective — ways of accomplishing this goal.

The best way to reduce the complexity of home navigation is to break it down into small easier to handle tasks. **Figure 2** shows a sample floor plan to demonstrate this idea. For this example, there are five places in the house that we would like the robot to be able to visit. Each of these places is designated by a numbered circle.

Let's start by assuming that there will be subroutines that can direct the robot from its current position to any

other position in the house. If the robot is at Position 1, for example, it could move to Position 3 by executing the subroutine *From1to3*. Similarly, if it was at Position 2 and wanted to move to Position 4, it could execute the subroutine *From2to4*.

Of course, each of these routines have to be created, but that is a much more manageable task. In order to help clarify this process, let's create the routine *From4to2* to move the robot from the Kitchen to Bedroom 2. To make things easier, let's assume that the top of the floor plan is north. Actually, this is not much of an assumption because the low level control of motors and sensors of Arlo is handled by the RobotBASIC Robot Operating System (RROS) which has a command to automatically add a fixed amount to all compass readings, allowing any direction to be considered north. With that in mind, imagine what has to be done in *From4to2*.

To move from the Kitchen to Bedroom 2, the robot

must turn west and move forward a small amount to ensure that it is adjacent to the wall (or the kitchen cabinet). Then, the robot should follow the wall until it finds a doorway. At that point, it simply moves through the doorway into the bedroom.

If we assume we have subroutines that perform the more complex actions, *From4to2* can be written as shown in **Figure 3**. Your first impression might be that this routine is larger than you expected from the short description above. There are several things that contribute to this complexity.

First, this routine was written to work with both Arlo and RobotBASIC's integrated simulated robot. Since both robots share similar sensor configurations, you can prototype many of the navigation routines on the simulator — especially if you create a floor plan of your home scaled as two pixels per inch. Doing so can greatly reduce the overall development time for many projects.

As we will see, many simulated behaviors can be used to control Arlo without modification, and many others often only need minor tweaking. As we examine **Figure 3**, the nature of such tweaking will become more obvious.

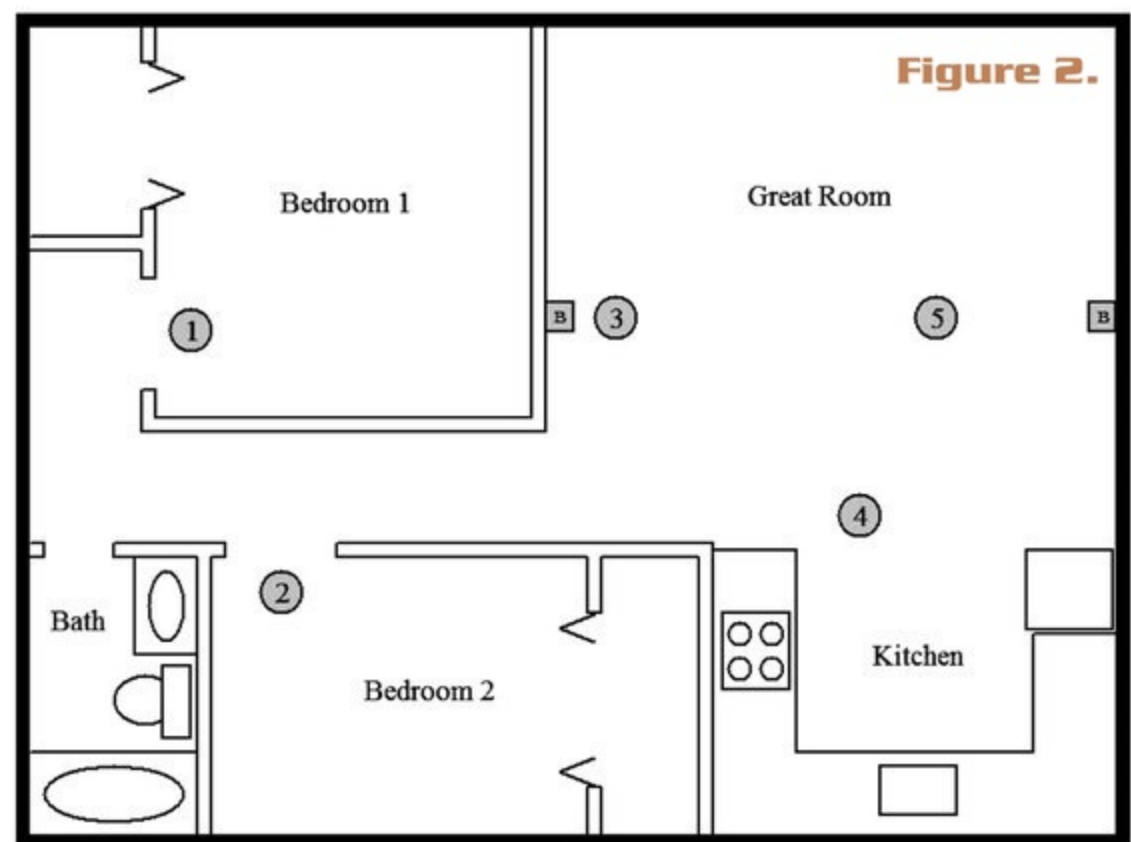
The first line in **Figure 3** returns from the routine without executing it if the variable *Lost* is TRUE. This is a way of letting the robot fail gracefully. Hopefully, Arlo never gets confused or loses its bearings due to an erroneous sensor reading — proper programming should certainly try to prevent such problems.

However, as anyone that has programmed a robot knows, unforeseen problems do occur. To lessen the impact of such situations, the robot's code should periodically perform checks to see if the robot at least appears to be where it should be; if it is not, the variable *Lost* should be set to TRUE.

Whenever *Lost* is TRUE, the robot should cease trying to accomplish the task at hand and perform some appropriate action. What is appropriate depends on the situation and your preferences. It could just stop and print a message on the screen or use its voice to call for help. A more advanced program might try to determine what caused the error in order to recover on its own.

The important point is that statements like the first line in the routine of **Figure 3** need to be added throughout the system at appropriate places.

If the robot is not lost, then the rest of the routine needs to move the robot from its current location (#4, the Kitchen) to the desired location (#2, Bedroom 2). As mentioned earlier, the robot needs to face west and move over to the wall before trying to follow it. Calls to three routines take care



of these actions. Details of what happens in these routines will follow, but if you are unfamiliar with RobotBASIC, a short discussion of the call itself could be beneficial.

RobotBASIC has two different types of subroutines. *From4to2* is a standard BASIC subroutine that is executed with a *gosub* statement. In addition, RobotBASIC offers subroutines that are executed with a *call* statement. These subroutines have the advantages of local variables and passed parameters. Additional details on subroutines will be given as we proceed. Let's return to our discussion of **Figure 3**.

After following the wall, the robot should already be facing generally to the west, but probably not a perfectly

```

From4to2:
  if Lost then return
  call Face(WEST)
  call FindWall()
  call FollowWall(OnLeft)
  if Real
    call Face(WEST+20)
  else
    call Face(WEST)
  endif
  if Real
    rForward 10 // Arlo measures from edge of robot
  else
    rForward 40 // simulator measures from center of robot
  endif
  if Real
    call MoveThroughDoor(SOUTH+15)
  else
    call MoveThroughDoor(SOUTH)
  endif
  call Face(135)
  delay 1000
  Cur = 2
return

```

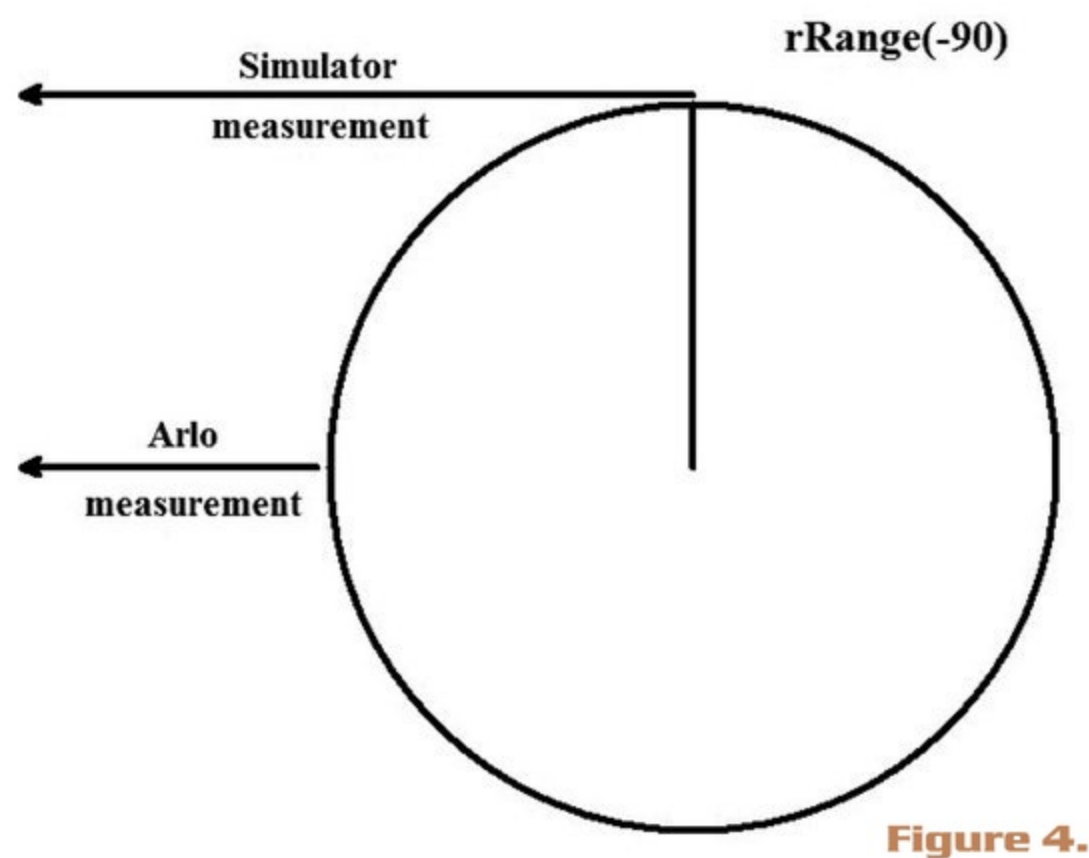



Figure 4.

```
sub Face(dir)
if _Lost then return
if _Real
  rCommand(_TurnToHalfAngle, dir/2)
else
  // turn in the shortest direction
  c=rCompass()
  if (dir>c) and (abs(dir-c)<=180)
    d=1
  else
    d=-1
  endif
  while rCompass()!=dir
    rTurn d
  wend
endif
return
```

Figure 5.

true heading. For that reason, another call will ensure the robot is angled properly so it can move forward to get in front of the doorway. This is a good time to examine one example of why some programs that work on the simulator need to be modified to work with a real robot in the real world.

Even though the RROS was set to synchronize Arlo's real compass with that of the simulator, real world testing showed that compass headings can be skewed when the robot is close to metal objects in the house (refrigerator, ductwork, etc). Luckily, even when such readings are wrong, they are generally repeatable. This simply means that the real robot might have to turn to a slightly different heading than the simulation at particular places in the home. You can see how this was handled in **Figure 3** when the *Face* subroutine is called to command Arlo to face in a specified direction.

Once the robot is at the end of the wall, the robot should move forward slightly to position itself in front of the door opening. This movement also needs to be different for real-world Arlo and the simulated robot. The simulated robot is assumed to have a turret at the center of its front edge. The turret holds a ranging sensor that generates the readings obtained by *rRange()*.

The RROS does support such a turret for real robots, but Arlo was configured to automatically use the reading from one of the PING))) sensors mounted around its perimeter to generate the *rRange()* parameter. This means that distance readings to a wall to the left of the robot will be longer for the simulator than for Arlo by an amount equal to the radius of the robot, as shown in **Figure 4**.

Since Arlo's range readings are units of a half inch, both Arlo and the simulated robot have an approximate radius of 20 units (the simulated robot has a radius of 20 pixels). This means a wall to the left of the robot by a

distance equal to the robot's radius will be reported as 20 for Arlo and 40 for the simulation, even though both robots are using nearly identical units of measure. Notice also that measurement made in the forward direction with *rRange(0)* will be the same for both robots.

The fact that the readings are taken from different points causes still another problem. The *FollowWall* routine called in **Figure 3** follows a wall until the range reading from the appropriate side (either right or left) increases quickly, meaning the wall has ended. This means the simulator will stop following the wall when its front nose protrudes past the end of the wall.

Arlo, on the other hand, will not see the end of the wall until its side sensor travels past the wall. Furthermore, since Arlo's PING))) sensors are ultrasonic they have a fairly wide beam, making the wall visible even longer. This means that Arlo will terminate its wall following activity only after the robot has moved most of its body past the final edge of the wall.

The whole point is that when the robot (either Arlo or the simulation) stops at the edge of a doorway, the front edge of the simulation will be barely in the doorway, while Arlo's body will be almost entirely in the door opening. This simply means we need to move Arlo an amount different from the simulation to approximately center it in the doorway. This is accomplished with the *if-else-endif* statement in **Figure 3**. Understanding the differences between the simulation and robots that you might create makes it easy to modify simulator behaviors so they work with your robot.

Once the robot is positioned in front of the door opening, a call to *MoveThroughDoor* moves the robot through the doorway, into the room where the robot is then turned to face the center of the room. Once you understand the overview of the move offered by the

routine *From4to2*, we can examine the details of each of the subroutines that do all the work.

The *Face* subroutine shown in **Figure 5** allows you to command the robot to turn to a specified heading. It starts with a check to see if the robot is lost (as discussed earlier). If the simulation is being used, the robot simply rotates in an appropriate direction until the *compass()* function indicates the desired heading has been reached. The real robot could do the same thing, but it would have to move extremely slowly to avoid overshooting the desired heading because of the delays associated with the Bluetooth communication between RobotBASIC and the RROS board.

To solve problems like this, RROS has special *rCommands()* that carry out time-sensitive functions entirely within the RROS itself. **Figure 5** shows how *rCommand* can turn Arlo to a specified heading. The angle specified must be divided by two so that a 0-360 degree argument can be passed as an eight-bit parameter. This does limit the resolution to 2° which is still better than 1%.

There is one more item that needs mentioning. The variables *Lost* and *Real* are global variables of the root program. To access them from a subroutine, the variable name must be preceded by the underscore character.

The *FindWall* routine in **Figure 6** allows the robot to move from a position (like Position 4 in **Figure 2**) over to the wall, so that a wall following behavior can begin. You might think you could simply move the robot with an *rForward* command, but this can cause problems if the robot is not exactly where you think it is.

For that reason, this routine uses Arlo's PING))) sensors (read with the *rFeel* command) to constantly adjust the robot's orientation as it moves in order to avoid colliding with the corner of the wall. Notice also that the maximum distance moved is different for real and simulated robots. The distance the real robot moves for an *rForward 1* depends on the robot's speed and the time needed to complete the loop, so it must be determined experimentally.

The robot terminates this behavior early when the front feel sensor is triggered, indicating the robot might be confused (lost) or when the left or right sensor indicates that a wall has been reached.

The subroutine for following a wall is shown in **Figure 7**. You must pass it *OnLeft* or *OnRight* so it knows where the wall is. The algorithm is relatively simple and is housed in the main *while-loop*. The robot records its current distance from the wall and stores it in the variable *r1*. The robot then moves a very short distance while monitoring its diagonal *feel* sensors to prevent a collision. At the end of the short movement, the distance to the wall is again measured and stored in *r2*.

These distances are used to control the robot's behavior, with priority given to keeping the robot parallel

```
sub FindWall()
  if _Lost then return
  if _Real
    dist = 40
  else
    dist = 55
  endif
  for i= 1 to dist
    rForward 1
    s = rFeel()
    // avoid corner of wall
    if s&2 then rTurn -1
    if s&8 then rTurn 1
    if s&17 // wall detected on left or right
      rForward 5
      break
    endif
    if s&4 then _Lost = True \ break
  next
return
```

Figure 6.

Arlo
Robotic Platform
SYSTEM

Build a
robust
custom
mid-size
robot!



PARALLAX
www.parallax.com


```

sub FollowWall(side)
  if side = _OnLeft
    Direction = -90
    SensorMask = 8
    REVERSE = 1
  else
    Direction = 90
    SensorMask = 2
    REVERSE = -1
  endif
  if _Real
    WallDist = 18
  else
    WallDist = 38
  endif
  while TRUE // main routine for following wall
    if _Lost then return
    r1=rRange(Direction)
    for i=1 to 10 // edit based on your robot's speed
      if rFeel()&4 then _Lost=True\ break;
      rForward 1
      while (rFeel()&SensorMask) // watch for wall
        rTurn REVERSE
      wend
      r2=rRange(Direction)
    next
    if r2>45 then break; // EOW
    change = r2-r1
    if change>0
      // first prioity - stay parallel with wall
      for j= 1 to 2+abs(change)
        rTurn -sign(change)*REVERSE
      next
    else
      // robot is parallel-try to find correct dist from wall
      for k=1 to 2
        if r2>18 then rTurn -REVERSE
        if r2<18 then rTurn REVERSE
      next
    endif
  wend
return

```

Figure 7.

with the wall (meaning *r1* and *r2* should be equal).

When the robot is not tracking parallel, the robot turns an amount proportional to the error. Once the robot is maintaining its distance, it moves slowly toward or away from the wall until the desired distance from the wall has been established.

This routine must allow either the real or simulated robot to follow walls on either the left or right side of itself, so the beginning of this routine sets up the specific values needed.

The *MoveThroughDoor* subroutine is shown in **Figure 8**. Once the robot faces the doorway, it simply moves forward while correcting its orientation to avoid the door casings, based on readings from the feel sensors. It is worth noting that RROS allows you to set the detection distance of the feel sensors.

Setting the feel distance so that the total width of the sensing area is slightly larger than the door's width can make traversing the door easier. Notice that the robot assumes it is lost if it does not appear to be at the doorway once it has turned to face it (something is directly in front of the robot, perhaps the door is closed).

An advantage of RobotBASIC's standard subroutines is that they can be executed using computed labels. This is a powerful concept that can be used as shown in **Figure 9**. The ability to compute the destination for a *gosub* statement makes it easy to direct the robot from its current position to any other.

The user could, for example, use the keyboard to enter a destination number, or touch the destination room on Arlo's touch screen.

A more advanced program might use a voice command to specify the desired destination as discussed in an article in the March 2014 issue of *SERVO*. The point is that no matter how the destination is specified, a computed *gosub* statement can easily execute the correct routine to direct the robot's movements.

Figure 10 shows the main and initialization routines needed to test a small portion of the navigational system on either the simulation or a real robot. Using a loop to move the robot continually back and forth between two locations makes it easy to find flaws in the behaviors used.

Due to space limitations, the subroutine *From2to4* is not shown, but it is included in

the code download available at the article link.

Normally, initialization of a real RROS-based robot requires many commands to specify necessary parameters such as the types of motors and sensors being used. Since Arlo is a standard platform, initializing using the current RROS version requires only two commands as shown in the *InitRobot* routine.

The *rLocate(ARLO,0)* sets up everything except the compass because specifying a non-existent compass can cause RROS to hang.

Building a navigational system can be a large and daunting task, but it can be greatly simplified by using the techniques discussed here. There are two important points to remember. First, the robot should not move any significant distance without utilizing some form of sensory feedback. In this article, for example, the robot followed walls when possible and used the compass and other sensors to ensure it could find a wall or its way through a doorway. In addition, the robot should periodically utilize sensor readings to determine if it has become confused or lost its way.

The routines discussed here only used the robot's perimeter sensors and its compass. Remember though, Arlo has both an IR and an ultrasonic ranging sensor mounted on the turret below its head. Utilizing these additional sensors gives the robot many more options, such as following a wall lined with furniture. The turret also has a beacon detector that can be used for unique situations. For example, suppose we wanted to create the subroutines *From3to5* and *From5to3*.

If you look back at **Figure 2**, it is apparent that there are no walls to follow as the robot moves between locations 3 and 5.

Since RROS can detect beacons (refer to the RROS manual; downloadable from www.RobotBASIC.org), two beacons could be positioned as shown in **Figure 2** (labeled B in the figure). The robot could then track to the relevant beacon until it is a specified distance from a wall. Techniques such as this can allow your robot to move to many more locations.

A robot could even move freely around the room and then use a beacon mounted above the doorway to find its way back to a known position. The angle to strategically placed beacons can also be used to help determine if the robot is lost.

As you can see, Arlo is an extremely capable platform, truly making it the robot you've always wanted.

Next month's article will discuss how to utilize Arlo's arms and turret to do even more things to impress your friends. **SV**

```
sub MoveThroughDoor(Direction)
  if _Lost then return
  call Face(Direction)
  while true // assumes rFeel dist is wider than the door
    if rFeel()&4 then Lost = True \break
    if rFeel()&2 then rTurn -2
    if rFeel()&8 then rTurn 2
    if rFeel()&17 then break //doorway found by left or right sensor
    rForward 1
  wend
  if _Real
    rforward 40
  else
    rForward 60
  endif
return
```

Figure 8.

```
Cur = 4 // current position
Dest = 2 // desired destination
gosub "From"+ToString(Cur)+"to"+ToString(Dest)
```

Figure 9.

```
#include "RROScommands.bas"
gosub InitCommands
Real = FALSE // set to TRUE to control the real robot

main:
  gosub InitRobot
  Cur = 4
  while not Lost
    Dest = 2
    gosub "From"+ToString(Cur)+"to"+ToString(Dest)
    Dest = 4
    gosub "From"+ToString(Cur)+"to"+ToString(Dest)
  wend
  xyString 500,300,"Robot is LOST"
end

InitRobot:
  if Real
    rLocate ARLO,0
    rCommand (SetCompass, HMC6352)
  else
    gosub DrawFloorPlan
    rLocate 580,350
    rSlip 10 // creates 10% random error
  endif
  OnLeft=1
  OnRight=2
  Lost = False
  Cur = 4 // specifies robot's initial position
return
```

Figure 10.

Using Desktop Machine Tools to Build Your Next Robot

By Gordon McComb

Post comments on this article at
www.servomagazine.com/index.php/magazine/article/march2015_McComb.

Among the types of machine tools in use, the mill and lathe top the list in both function and versatility. Both have been used in manufacturing goods for decades; in fact, the industrial economies throughout the world owe much of their success to the mill and lathe. Historically, both have been leviathan in size, weighing more than the average adult, and some requiring special electrical connections.

Enter the desktop machine tool.

These tools are similar in features and function to the ones you might see at large machine shops, but are scaled down in both size and price — an ideal mix for amateur and educational robot makers. Model makers have used desktop mills and lathes for years, but are relatively new to the robot builder.

Also important, the latest desktop machine tool either comes equipped for or is readily adaptable to control via a computer. With a CNC (computer numerically controlled) lathe or mill, you can load a design file into a PC, and have most or all of the complex machining done for you.

In this article, we'll discuss the use of desktop machine tools in a bit more detail. You'll discover what's available and what you can do with it. Being small versions of industrial mills and lathes, the desktop machines are obviously not meant to build a new engine for your Chevy. However, they're perfect for small scale robotics.

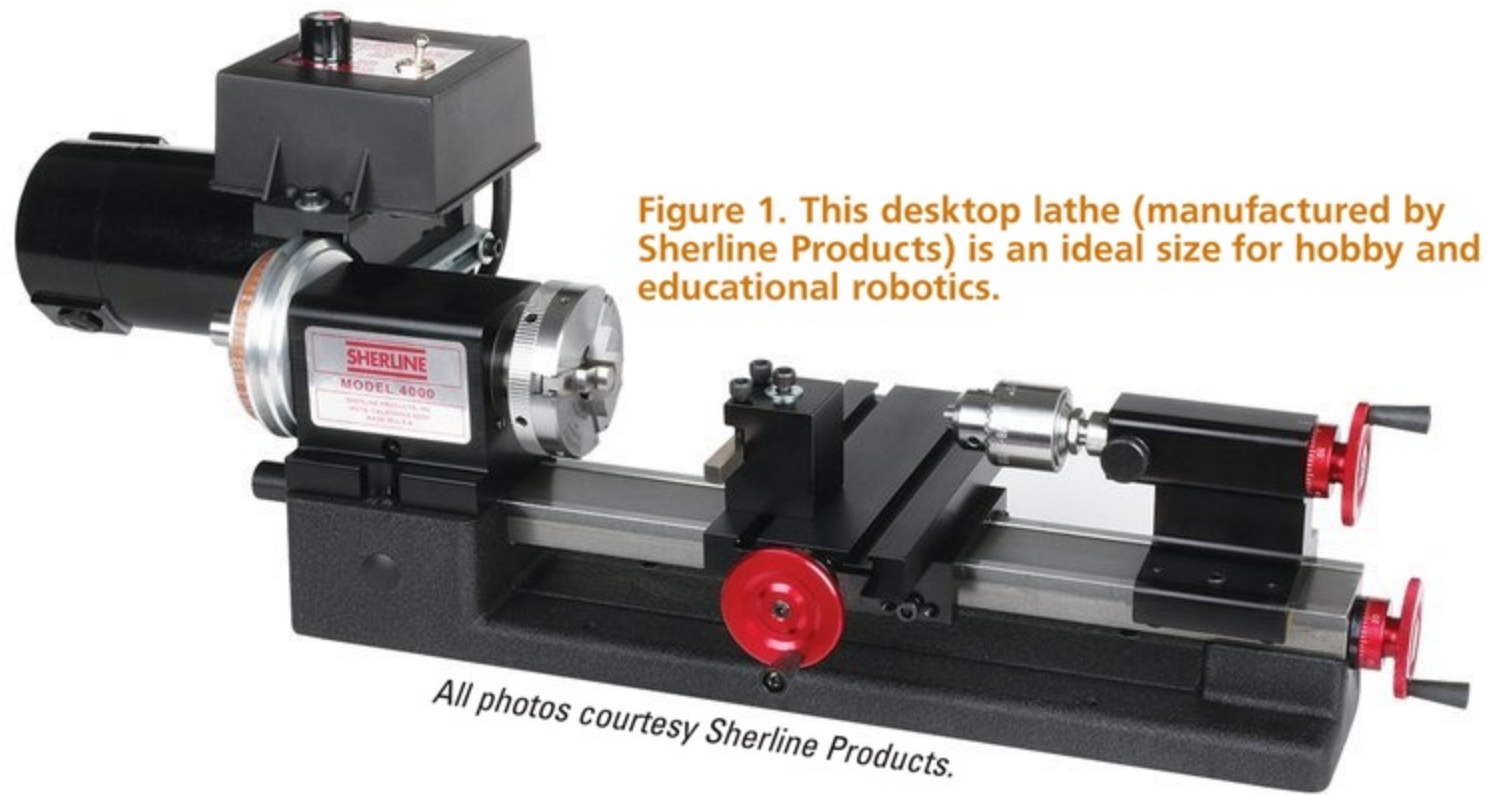


Figure 1. This desktop lathe (manufactured by Sherline Products) is an ideal size for hobby and educational robotics.

All photos courtesy Sherline Products.

So, let's begin.

What's a Desktop Machine Tool?

First, some definitions:

- A *lathe* is used to rotate a part against a cutting tool. It is typically used to contour square or cylindrical material into some useful shape — examples include specially-made bushings or threads on a rod. In the most common arrangement, the material you're cutting turns on a horizontal bed; a movable (but non-spinning) cutting tool is brought up against the material to carve it out into the desired shape.

Figure 1 shows a popular desktop lathe from Sherline Products.

- A *mill* looks a lot like a vertical drill press. In fact, that's basically what it is. Although, the typical desktop mill is built for (far) greater accuracy and control. Instead of a drilling bit that just goes up and down, on a mill the work piece itself can be moved horizontally and laterally. This allows the mill to cut out complex shapes instead of just drill holes.

Materials You can Use with a Desktop Machine

Because desktop mills and lathes use very sharp cutting tools, you can

Even a simple robot needs basic construction tools. The more elaborate the robot — and more specialized its building materials — the more sophisticated the tools you may require. In addition to the saws, drills, wrenches, screwdrivers, and other common tools of the robot building trade, there's also the more esoteric: machine tools capable of producing highly accurate and detailed small parts.

use them to create your robot parts from all kinds of construction materials. Aluminum is a common choice when you want the utmost in strength. It comes in sheets of various thicknesses, bars, tubes, and rods. **Figure 2** shows a “perpetual motion machine” project created on a desktop mill out of a sheet of aluminum.

You never really work with raw aluminum, but rather an *alloy* that mixes aluminum with other materials (commonly magnesium and silicon) to improve the performance of the metal. A common aluminum alloy for machining is 6061. The aluminum is often treated using heat or various chemical solutions to change its properties.

For example, 6061-T6 refers to an alloy of aluminum that has been *tempered* to make it even stronger. A drawback to tempered alloys is that they are often harder to cut and drill, so the process of machining with them takes longer.

While aluminum is a common material for desktop machining, it's by far not the only one you can use. You can opt for other lightweight metals like brass, tin, or copper. Use these other metals when you want to give your robot a distinctive look. For example, polished brass can give your bot a steampunk appearance.

In addition to aluminum, materials such as stainless steel and titanium can be cut, as well. The only difference between cutting materials on a desktop machine versus a full-size machine is the amount of material (depth of cut) and the rate of material removal (feed rate).

On a desktop machine, the depth of cut and feed rate must be lower because you are working on a smaller machine.

Selected Makers of Desktop Machinery

There are a number of makers of desktop mills, lathes, and CNC routers. Among the most popular manufacturers of desktop mills and lathes are:

Flashcut CNC
www.flashcutcnc.com

MAXNC
www.maxnc.net

Minitech
www.minitech.com

Sherline
www.sherline.com

Taig
www.taigtools.com

Prices and features from these makers vary. If you are interested in acquiring a desktop mill or lathe, you're well advised to get information on as many of them as possible. The typical starting price for a better-made tool is \$500, so consider research part of your investment. Keep in mind that not all desktop tools are created equally. Price goes up based on accuracy, power, size, and warranty, so plan your purchase accordingly.

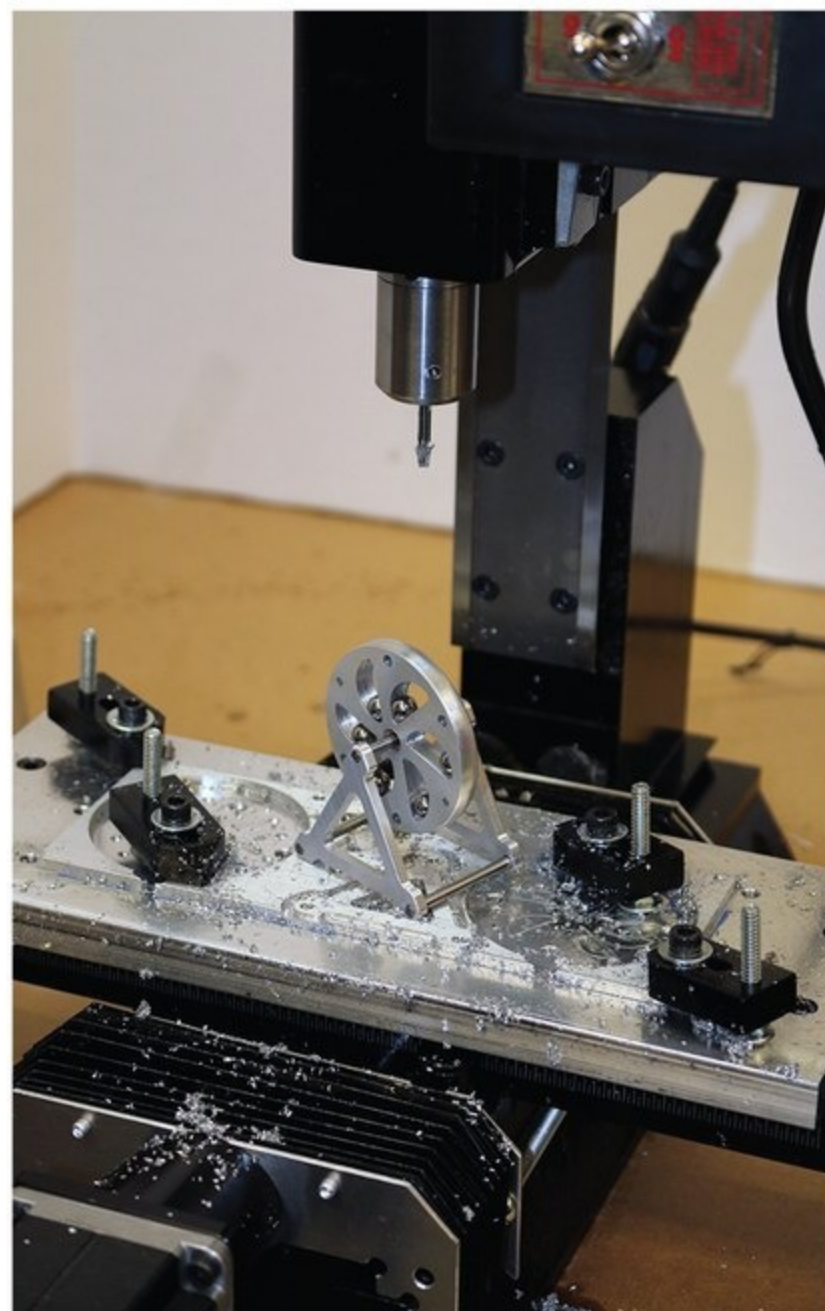


Figure 2. Aluminum is a prime material to use with a desktop lathe or mill. Parts can be cut from an aluminum sheet to make various types of projects.

Harder materials can be turned or milled on desktop machines when using the proper tools (carbide, insert tooling, and ceramic inserts).

There's also numerous plastics, some of which — like nylon, acetal (Delrin), and ABS — are perfect for making stuff using a machine tool. Plastics tend to be softer than metal, so they cut faster. For many applications that don't require tremendous strength, plastic is often the better choice because you can make your parts more quickly, and your cutting tools don't wear down as fast.

Finally, don't forget wood. Though machine tools are most often associated with making parts out of metal, for a small robot wood is still a

perfectly acceptable alternative. This is especially true when using the harder woods — oak or ash are readily available in different sizes and shapes, reasonably priced, and are modestly hard. Small lightweight robot parts can be quickly machined from pine and fir and other soft woods commonly available at your local home improvement store.

Of Accuracy and Precision

At the heart of both the lathe and mill is their ability to precisely drill, cut, and shape parts. Handwheels, dials, verniers, and accurate leadscrews on

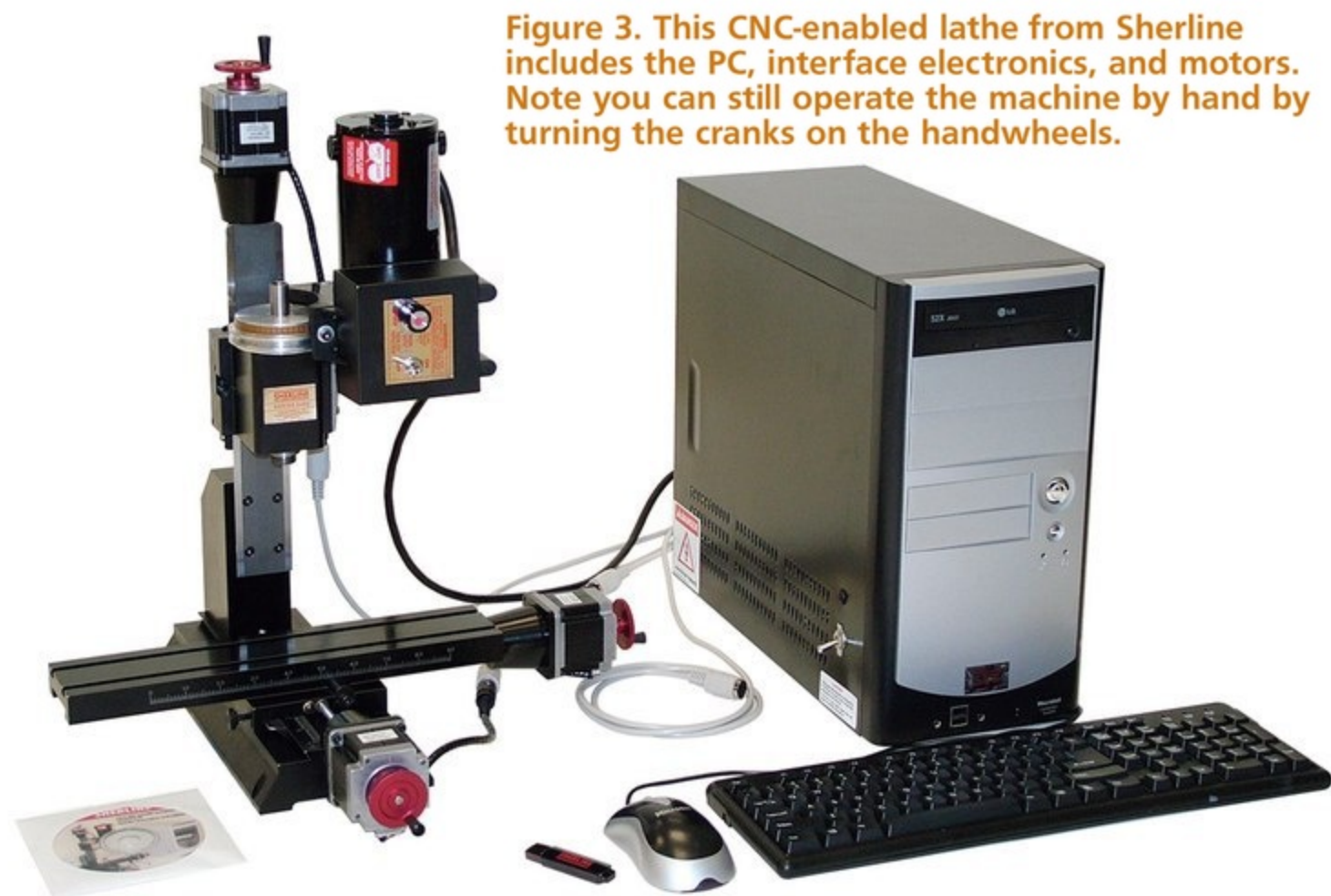


Figure 3. This CNC-enabled lathe from Sherline includes the PC, interface electronics, and motors. Note you can still operate the machine by hand by turning the cranks on the handwheels.

you. Your PC reads a set of cutting instructions and operates the motors accordingly.

You can either purchase a desktop mill or lathe with the CNC parts already attached, or upgrade an existing tool by adding the required stepper motors, interface electronics, and software. Many sellers of machine tools such as Sherline offer retrofits that you can add yourself (see **Figure 3**).

At the heart of any CNC machine is the software used for controlling the motors. This software reads a standard computer aided design (CAD) file and converts it to incremental motions for each of the motors attached to the machine.

A standardized language called G-Code is most commonly used for CNC control. G-Code is human readable — the instructions are in plain text and can be modified as needed using any text editor. An example of a G-Code program being run on a CNC machine is shown in **Figure 4**.

Two Added Benefits of a CNC-Driven Lathe or Mill

The machine can run more or less independently, allowing you to do other work. How much you need to keep your eyes on the machining process depends on what you're fabricating. When using aluminum, for example, you should periodically add cutting oil to keep things running smoothly. (You can avoid this by installing an automatic oiler to your mill or lathe.) You may also need to periodically stop the machine to change the cutting bit to another size or type.

It's easier to make clones of your parts when using CNC. The computer removes the human error that can occur when manually operating a mill or lathe, and makes for more reproducible parts.

That's the good news about CNC machine tools; now for the bad. A full CNC package for a mill or lathe — including a dedicated computer and software — can add several thousand

hands-on practice. Be prepared to spend numerous weekends just playing with your machine tools. Before tackling a full robot construction project, try making a few simple shapes. Look around the Web for simple projects you can try. These projects are most commonly presented as a drawn plan which you should print out and keep beside you as you create the project. Part of the process of learning to use machine

tools is studying how to read machining plans. In time, you'll be able to grind out even sophisticated parts.

Using a Computer to Automate Drilling and Cutting

Not everyone has the time or inclination to become a master with their desktop mill or lathe. You can automate much of the machining process by adding CNC. Rather than manually turning the handwheels on the machine tool, your computer commands a set of motors to do it for

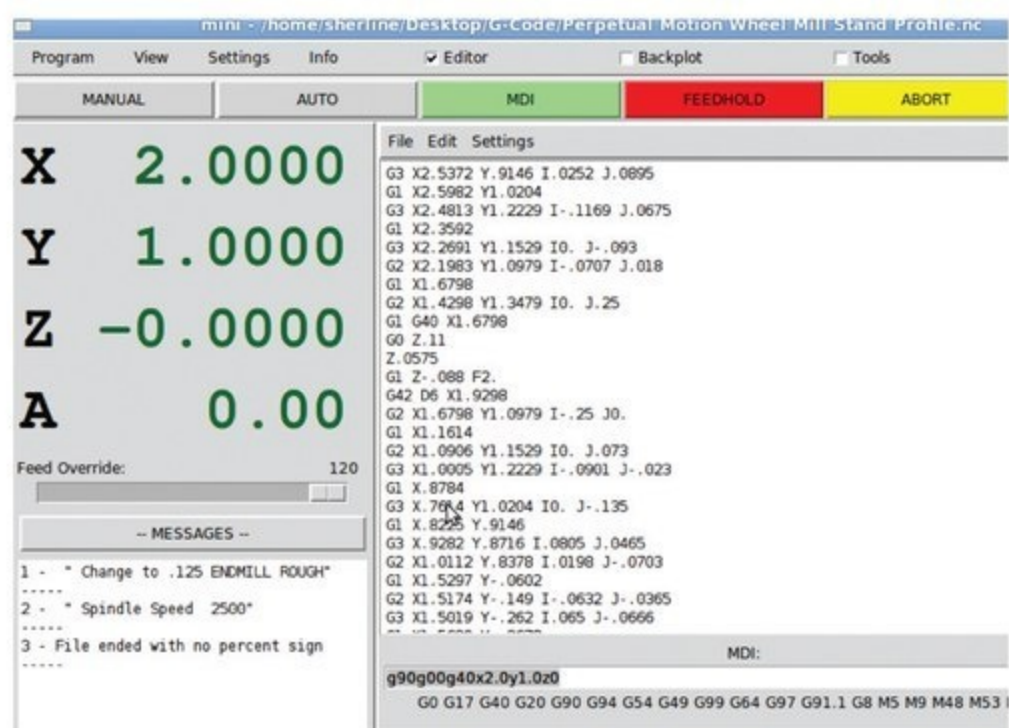


Figure 4. When using CNC, parts are cut by following a set of G-Code sequences like those shown here. While the coding syntax looks complex, it's actually fairly simple, and is composed of just a few basic commands.

the machine let you move the cutting tool precisely (or material being cut) in small increments at a time. Unlike using a drill press where you rotate a handle to move the drilling bit inches at a time, on a machine tool you move the cutter mere fractions of an inch. It's a more precise process that yields finished parts that are far more accurately cut and drilled than using ordinary shop tools.

While precision and accuracy are the hallmarks of machine tools, it takes practice to effectively use them. Basic concepts can be learned in an afternoon for the lathe, and maybe a day or two for the mill. Like anything, your skills are improved only with

dollars to your investment. You can reduce costs by using an existing PC, but not all modern computers are geared towards controlling stepper motors. Many systems rely on either the old parallel printer port for stepper motor control, or need a custom interface to bridge between the computer and stepper motors.

Try Before You Buy

Increasingly popular are "maker zones" where you can join for a nominal fee and rent time on various construction machines. Most such

businesses require you to either demonstrate a minimum level of competence with the machine or take a training class beforehand. Lathes and vertical mills are common finds at most maker zones, though the scale of the machine may vary. Some are equipped with full-sized machines, while others offer the smaller desktop units.

You can use these zones to try out a lathe or mill before investing in one yourself. Classes are a good way to both discover the personal benefits of the machine and practice while learning. Having an expert nearby to

answer questions is invaluable, and can cut down on your learning curve dramatically.

Finally, if you don't have one of these zones nearby, consider joining some Internet forums on desktop machining. Find some local groups where you can visit, observe the members in action, and ask questions.

Among the typical users of desktop mills and lathes are crafters, model railroad enthusiasts, and R/C airplane and racing buffs. Attend local shows and fairs in your area related to these hobbies, and make connections.

SV

A Brief Look at Other CNC Desktop Machining Tools

Desktop Laser Cutters

Growing in popularity over the last several years is the desktop laser cutter. These are computer-operated machines that use bright pinpoint laser light to cut sheet material.

Laser cutters differ in their size and light output. A small desktop laser may be limited to cutting material no larger than a writing paper; more practical models can cut bigger pieces up to 18" or 24" wide. Also, the power output of the laser determines the types of materials you can cut, the maximum thickness of the material, and the speed at which the material is cut.

CNC Routers

For the robotics hobbyist with extra spending cash, a CNC router will have you building all sorts of robots in record time. A CNC router combines a high speed motorized cutting tool (the router) with computer-controlled movement. This movement is controllable in three planes: X, Y, and Z.

CNC routers are inherently a computer-controlled device. Mills and lathes can be completely manual affairs, or they can be hooked up to a computer. With most models, you can purchase a manual desktop mill and lathe today, then sometime down the road retrofit it for computer control.

There are several benefits of CNC routers:

- They're designed to handle larger work pieces. A larger (but still "desktop") router might cut sheet material up to 2 x 3 feet. (However, note that with their increased size, CNC routers aren't generally as accurate for small pieces as desktop mills.)

- They can cut wood, plastic, thin metals, and other sheet materials. Favorite construction materials include aircraft-grade plywood and expanded PVC. The latter is commonly used in the sign-making industry. It's a lightweight material that shares many of the same benefits of pine and other soft woods.
- Because the machine uses a high speed router instead of heat — as is the case with a desktop laser cutter — you're not limited to the types of plastic you can cut. PVC is particularly onerous when cut on a laser (it emits a toxic gas), but poses no health hazard when cut on a CNC router.

To use a CNC router, you first secure a piece of wood, plastic, metal, foam, or other material to the base of the machine. As with all CNC tools, you then program the moves the router will take over the material. For example, if you're cutting out a shape, the router will first move downward (the Z axis) to drill into the material, then move back and forth (the X and Y axes) to cut out the shape.

Depending on the routing tool used in the machine, you might also be able to mill parts out of softer materials, like PVC or soft woods. (For metals and other hard materials, a mill is a better choice as it has more cutting power and accuracy.)

While CNC routers are simple in principle, they are not cheap in cost. A typical ready-made CNC router package for hobbyist use costs around \$2,000. Prices go up from there, with \$5,000 being typical for an entry-level professional model. An example of a medium size machine is the CNC Shark from online woodworking retailer, Rockler.

You might save some money building your own. Several online resources (such as Kronos Robotics) sell plans and parts to construct a homebrew CNC system. Whether or not this is an effective alternative for you depends on your experience level and construction skills.

The most expensive parts of any CNC system are the motors and control electronics. Most CNC routers use stepper motors which come in standard sizes and configurations. You can sometimes find these motors for less in the surplus market, but take this approach only if you have experience with stepper motors, their specifications, and their applications.

3D Printers

Quickly moving up the pop charts are 3D printers, which are strikingly similar to the CNC router. However, where a router makes stuff by drilling and cutting away excess material using a cutting tool, a 3D printer is "additive." It forms parts by coating on layer after layer of quick-hardening plastic ooze until the whole piece is fully formed.

As with CNC routers, 3D printers are inherently computer-controlled. A typical printer comes with control software that either includes a 3D design program of its own, or reads from a standard 3D file (STL and DWG are commonly used). To make a part, you load the plastic forming material (a thin plastic rope on a spool is fairly standard), load up the design you want to print, and click Go. Other technologies are used in some 3D printers, but this is the type you'll see most often.

3D printers vary greatly in their speed and maximum part size. Speed is a function of both the speed of the motors and the molding process. Of course, the larger the part, the longer it takes to make. Smaller printers may be limited to constructing parts under about four inches in height, width, and depth. Larger printers double that capacity, allowing you to form all or most of the body parts for a desktop mobile robot.

A core benefit of the 3D printer is all of the free design files available for them. Thingiverse (www.thingiverse.com) is a primary online source for all kinds of project files for 3D printing (they also have 2D files suitable for laser cutters and CNC routers). You can search for projects to make on your machine by category or keyword.

Unwrapping a da Vinci 3D Printer

By Steven Nelson

Post comments on this article at
www.servomagazine.com/index.php/magazine/article/march2015_Nelson.

It never ceases to amaze me the places I end up going to, the events I build for, and the people I've met since I got involved in robotic combat and robot building in 1997.

The first robot combat event I attended was Robot Wars 1997. It was held in San Francisco, CA at Fort Mason. My father, Lowell Nelson and I watched that event and video taped it from the stands. Attending that event planted the builder seed in us, and we decided to create our own combat robot.

Since we are both automotive/diesel mechanics, we figured we could adapt some sort of gasoline engine to power the weapon, and then add some sort of electric drive train to move the machine around. That was the beginning of our combat bot called S.L.A.M. That build started an endless search for information using this semi new thing called the Internet.

At the time, there was very little information available on how the robot combat machines worked. There were only a few web pages and very few pictures. So, we mostly relied on watching the video we took at Robot Wars over and over, and also searching the Internet for basic components like small engines, electric motors, batteries, motor speed controllers, radio controls, wheels, and lots of other stuff. Eventually, I got really good at using a search engine, and also ordered and read a lot of parts catalogs.

I emailed most of the builders I could find like Jim Smentowski (Team Nightmare) since he had a combat robot called Hercules. About this time, Carlo Bertocchini — the

builder of Biohazard — started a forum called Robot Wars on the Delphi forums website. Jim sent me an email and told me about the new forum, so I signed up. Eventually, several other builders learned about the forum, and we began sharing information about parts and our designs.

The Robot Wars forum was great since you could ask questions from the very few veteran builders and have access to everyone's knowledge base — if they would answer your questions without flaming you. While building S.L.A.M., I collected tons of information and began sharing my findings on the forums and my website to help folks out, and also to help make the sport of robotic combat grow. I started doing build reports, and talked about our successes and (more often) failures during the build/learning process. We did have a lot of failures, fires, and periodic bouts of human blood letting when a tool, torch, welder, grinder, human, or robot did something naughty or foolish. We often called our build process "trial and terror."

We did eventually compete with S.L.A.M.; first, at the Underground Robot Street Fight and then BattleBots. Around that time, the Robot Wars forum was forced to shut down for posting due to copyright issues with its name. So, several more forums such as the BattleBots, SORC, and RFL were started. The robotic combat community grew world wide.

Since then, I have built a lot of robots (61) and never stopped posting about them — or anything else that's related to science or technology. So, here we are 17 years later ...

An Unexpected Chat on Facebook

As usual, there I was reading the posts on the Robotics Community page on Facebook when I received an instant message from a fellow builder, Jerome Miles (Team Duct Tape). Through the forums, I have communicated with Jerome since the early days when he was about 14 years old, and we have met and competed at several events. So, Jerome asks me a question, "Hey buddy, do you currently own a 3D printer?" All I said was, "No." Jerome said, "That needs to be rectified. Would you rather have an Up Mini or



a da Vinci 1.0?" I told him I didn't know much about the different models. Jerome continued, "You can take some time to decide. Check those two models out and let me know which one you like better. I know it's out of the blue, but I've been thinking about you lately and I would like to purchase one for you. I think your projects will get exponentially cooler if you have one around, and I enjoy reading about them on Facebook."

I figured he was kidding, but then he told me, "Naw, you deserve it man. You and your dad helped shape me into who I am back in the old BattleBots days, and I would just like to pay you back." Still in shock, I told Jerome I'd get back with him, and then I spent the rest of that night on the Internet researching the two printers he had mentioned.

I was back on Facebook the next night when suddenly the chat window popped up. It was Jerome asking how my research was going. Basically, I told him that the XYZ da Vinci 1.0 looked a bit more newbie friendly than the other one. I gave him my address and choices for some extra ABS plastic filament colors. A few minutes later, he gave me the UPS tracking number.

Lots of Cool Stuff in the Mail

Two days later, I got a knock on the door. It was UPS with a box of ABS filament. When I was at work, I checked the tracking number and another box had been delivered. I raced home and there it was — a brand new da Vinci.

The box was so big and heavy, I barely got it through my doorway without grating all the flesh off my hands. I looked at the 50 year old two-prong power outlets in my house and realized I had to go buy some new three-prong outlets and install them to work with a computer and the new printer. When I started the unboxing process, it felt like it was Christmas times five!

Unboxing the 3D Printer

The unit was very well packaged and supported in the box with Styrofoam, so much so that it was actually a challenge to get it out of the box. This is a good thing,

though, since the shipping process can often be a rough ride for sensitive precision devices.

On top of the printer was a sheet of cardboard with all of the accessories attached to it with orange colored tape. The accessories include the printed unpacking/setup instructions, the side covers, a wire brush for cleaning plastic from the extruder, a CD with the software, a scraper, the power cord, the USB A-B cable, one UHU glue stick, and some small wires for cleaning the extruder nozzle. The printer also comes with one cartridge of ABS filament.

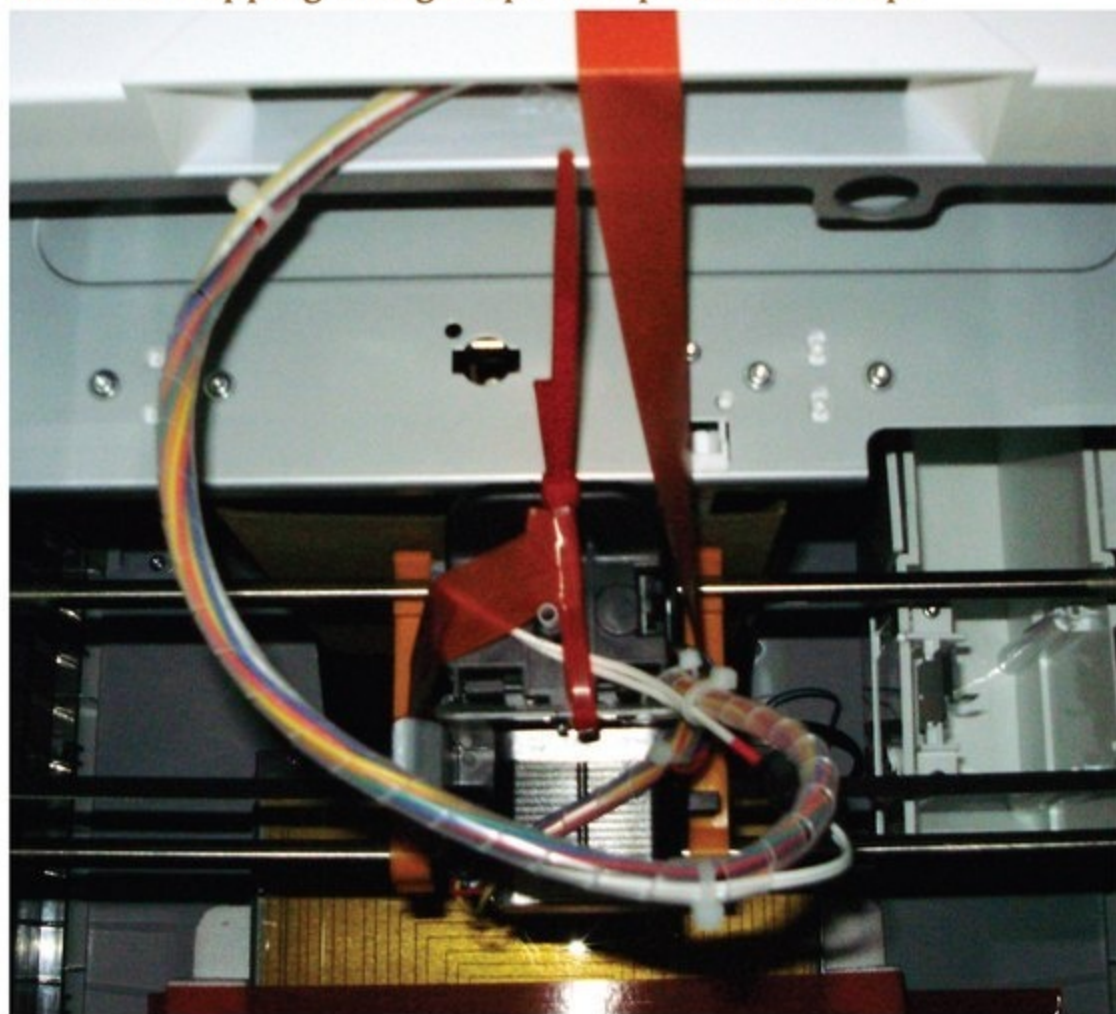
Once I got it out of the box and on the table, I read the instruction sheet and began removing the orange tape from the door. Once inside, you have to remove a lot more tape that immobilizes the belts and the extruder head. There's also a tie-wrap, plus two orange plastic shipping blocks holding the extruder. There's a large cardboard block, an orange plastic block, and Styrofoam blocks that immobilize the heated platform.

During my research, I read there were some problems with damage during shipping with some of the early printers that were sent out. Apparently, the manufacturer



The da Vinci accessories.

Extruder shipping orange tape and plastic tie wrap.



addressed this problem. You really need to read the instructions to get all of the tape and various restraints removed before you power the printer up. The next thing you have to do is install the software on your computer.

Installing the Software

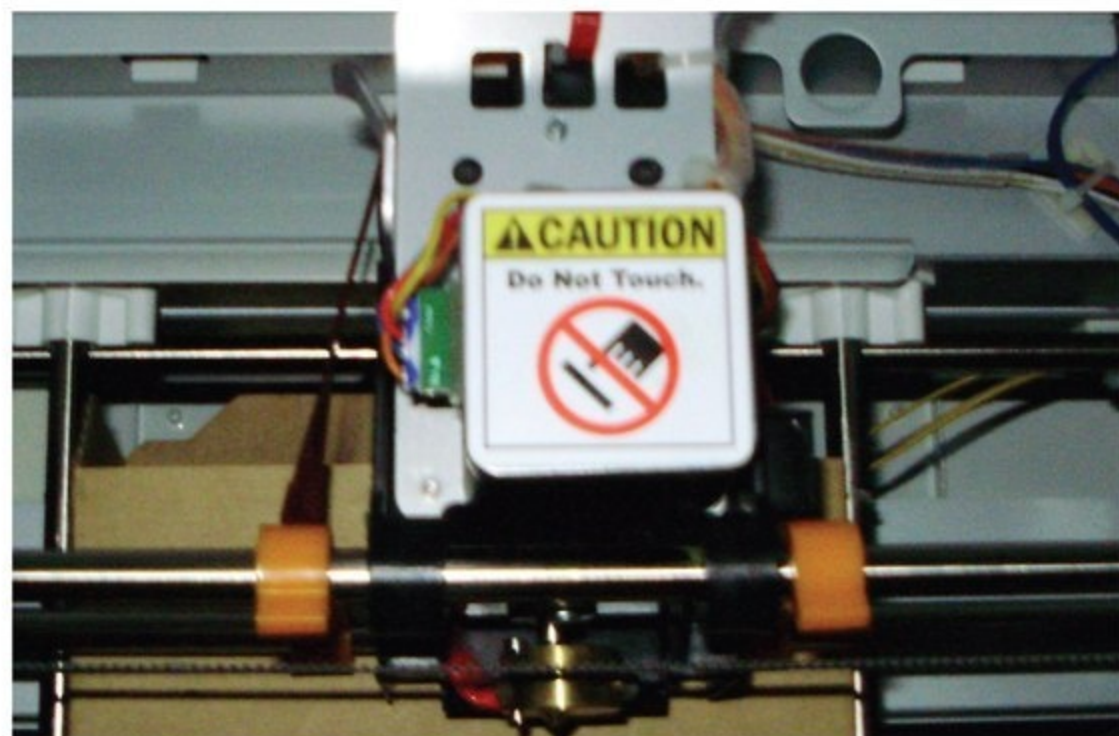
The software comes with XYZware for older versions of Windows (like XP or Windows 7), however, there is a note on the instruction manual that tells you to go to the XYZ website for other operating systems like the MAC OS or Windows 8/8.1. I needed the software for Windows 8.1. (Note: You may have to turn on permissions for adding third-party or unlicensed drivers in Windows 8/8.1. Fortunately, mine were already turned on.)

I downloaded and registered the printer with XYZ. The computer set up the software and loaded the drivers. I was then able to connect the USB cable and turn the printer's power switch. Windows 8.1 found the printer and installed the driver with no problems. That was nice. I then powered down the printer, so I could install the ABS plastic filament cartridge.

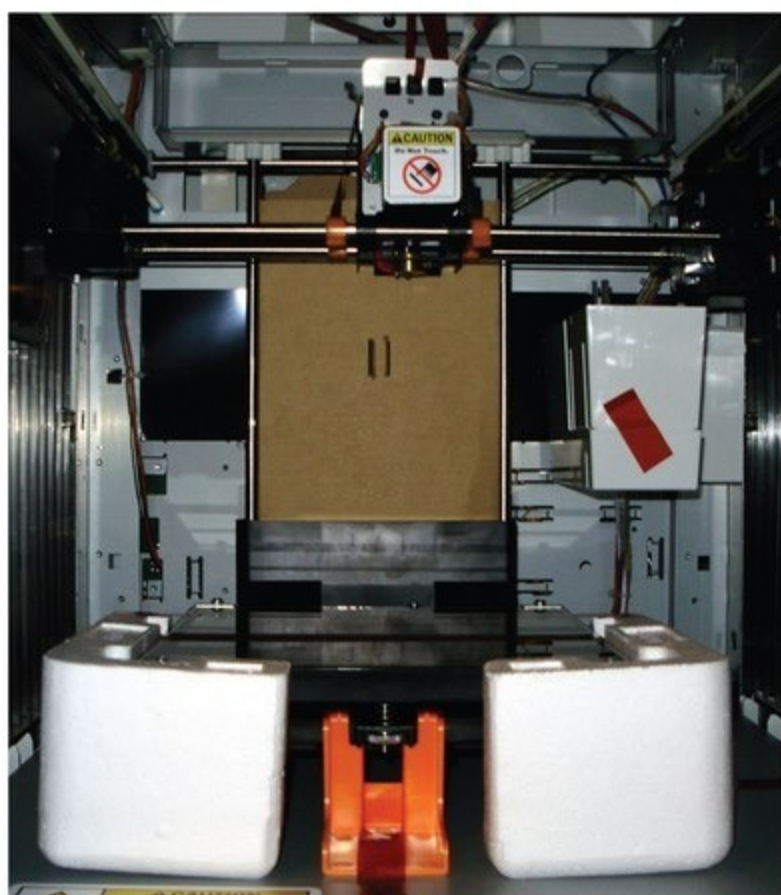
Installing the Filament

The da Vinci 1.0 3D printer uses proprietary cartridges that contain the ABS filament. So, it has its own procedure for installation. There is a plastic cartridge retainer block you remove and install into the rear of the printer housing. Before you install the cartridge, you have to remove a small rubber plug (filament stopper) from the cartridge housing. You can then just slide it in the machine.

Once it is in place, you remove yet another piece of tape that holds the loose end of the filament to the cartridge; cut it off at a 45 degree angle. Next, you feed a nice angled piece into a plastic guide tube, and then finally



Orange plastic extruder shipping blocks.



Shipping blocks for the heated platform.

All of the shipping blocks and orange tape removed.





ABS filament cartridge.

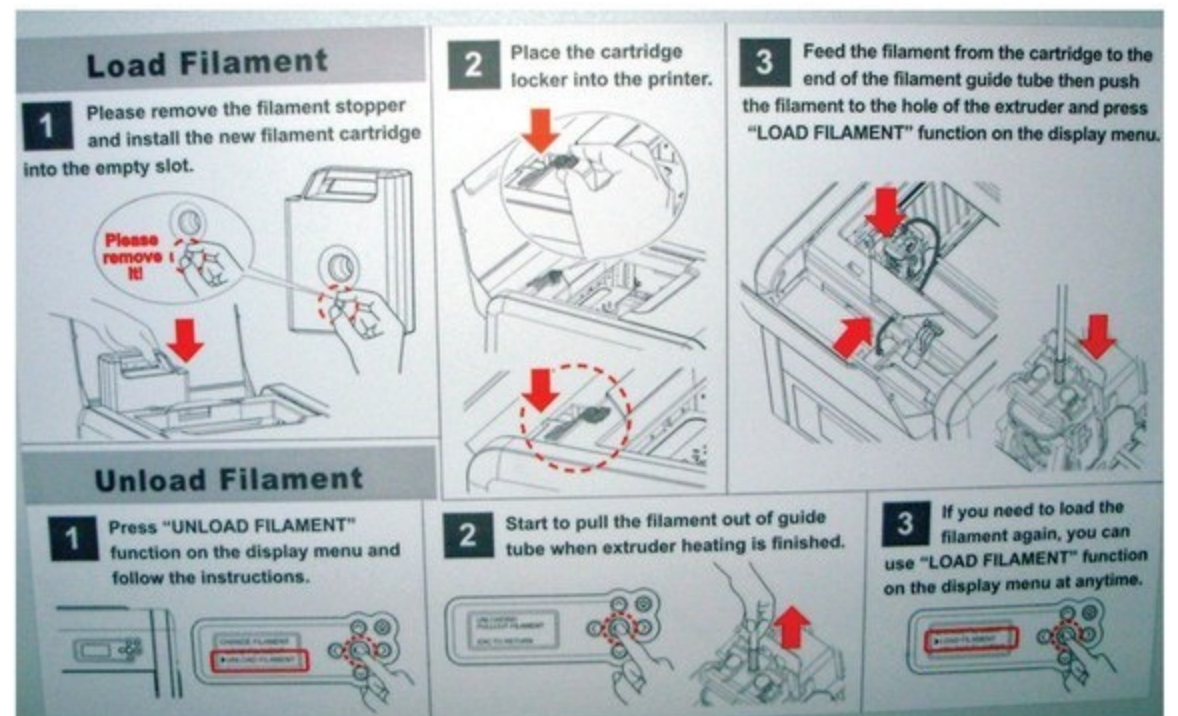
into the top of the extruder head. After that, you power the printer back up and go through a LOAD FILAMENT procedure using the menu on the LCM display on the front of the printer.

The printer will start heating the extruder. Once it is at its operating temperature, it will try to feed some filament. Well, of course it didn't at first, so I had to manually push the filament in by hand until it started flowing (apparently, I was just a bit *too* gentle when I loaded the filament in the extruder head).

Regardless, it started working. This whole process from



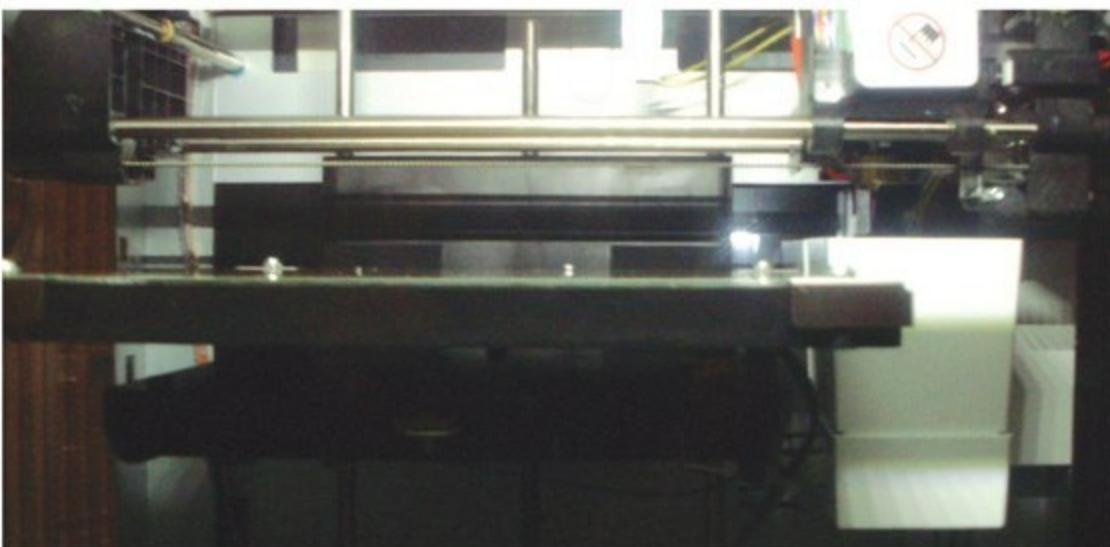
ABS filament stopper plug removed from cartridge.



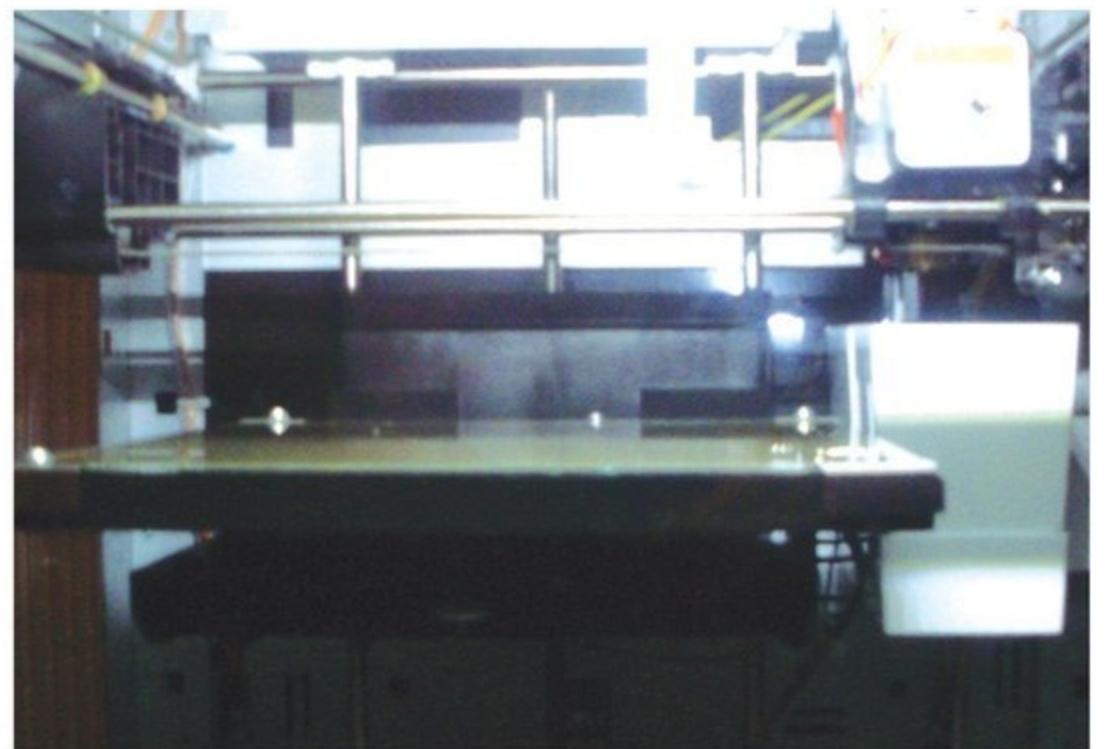
Loading the ABS cartridge instructions.



ABS filament cartridge installed with retainer block.



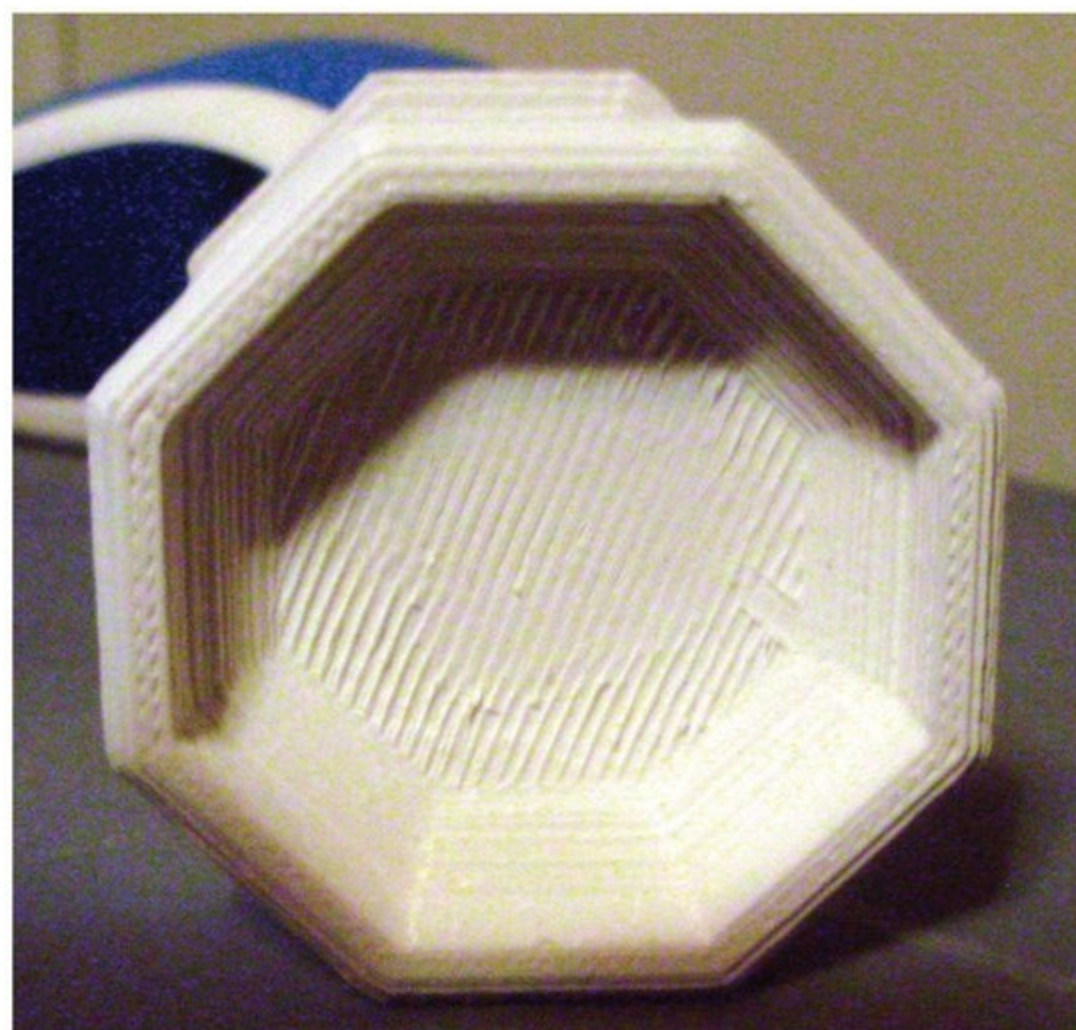
Heated platform almost at its home position.



Heated platform rising up to the extruder.



First
printed
object
from
"demo"
— the
chalice.



unboxing to loading the software to adding the filament actually took me about an hour or so, mainly because I was being very careful to read and follow all of the printed and online instructions.

My First Printed Object was a Surprise

The printer comes with three samples loaded into its internal memory: demo, key chain, and star vase. Before I printed anything, I did watch a couple YouTube videos about using the da Vinci printer and it turned out to be very helpful. I also learned what the UHU glue stick was for.

The printer uses a heated base that is covered with a sheet of glass; you have to coat it with UHU glue to get the ABS plastic to initially stick to it. So, I guessed where the center of the glass was and applied a four inch square of glue to that area. I then selected the demo design from the menu, and pressed OK. The printer said it was heating. It does take a few minutes for the printer base and the extruder to heat up, so you just have to wait. After the parts reach their operating temperature, the heated base begins to rise as its stepper motor and lead screw turns.

Eventually, the base gets to the top and automatically finds its Z axis home position. The other two stepper motors move the extruder head and actually clean its tip with the built-in scrapers. The steppers move the extruder to the edges on the heated plate and self-calibrate themselves for X-Y axis positioning.

Next, the motors move the extruder and print a couple of lines on the right side of the heated base. I wondered what that was for — it turns out this is normal, and it helps make sure the filament is flowing correctly. I also learned

that you need to put some glue in this area before printing, as well.

The machine actually dragged that small piece of filament to the center of the heated base when it moved there to print the actual object. I was puzzled that it printed a pink line since I had loaded white filament. I learned later that this is normal and it was just some leftover filament from the factory testing.

The demo print started and I was delighted to observe the stepper motors making a lot of sounds while doing their thing. It kind of sounds like techno music — something like the band from the Cantina Bar in the movie, *Star Wars*.

Having only briefly observed some of the early homebrewed 3D printers at a Maker Faire, I was fascinated watching this machine do its work. I had no idea what it was going to print in demo mode. I stared at it as it deposited layer after layer for about 30 minutes. Turned out the demo was a small chalice with a base, a fluted stem, and an octagon shaped cup on the top. After it was done printing, the display said, "Wait Cooling" until the platform cooled down and lowered itself to the bottom of its travel.

As a side note, there are a couple of warning stickers inside the machine that mention to watch your hands because of moving parts, and to allow the surface to cool before removing objects or servicing. (These guys must know me or at least have read some of my build/incident reports over the years.)

After the printing and cooling was completed, I used the included scraper to unstick the chalice from the platform. I marveled at the ability of 3D printing. I cleaned the platform, then reapplied some more UHU glue so I could print the star vase (which was also very cool to watch since it appeared as if by magic).

XYZ da Vinci 1.0 3D Printer Specs

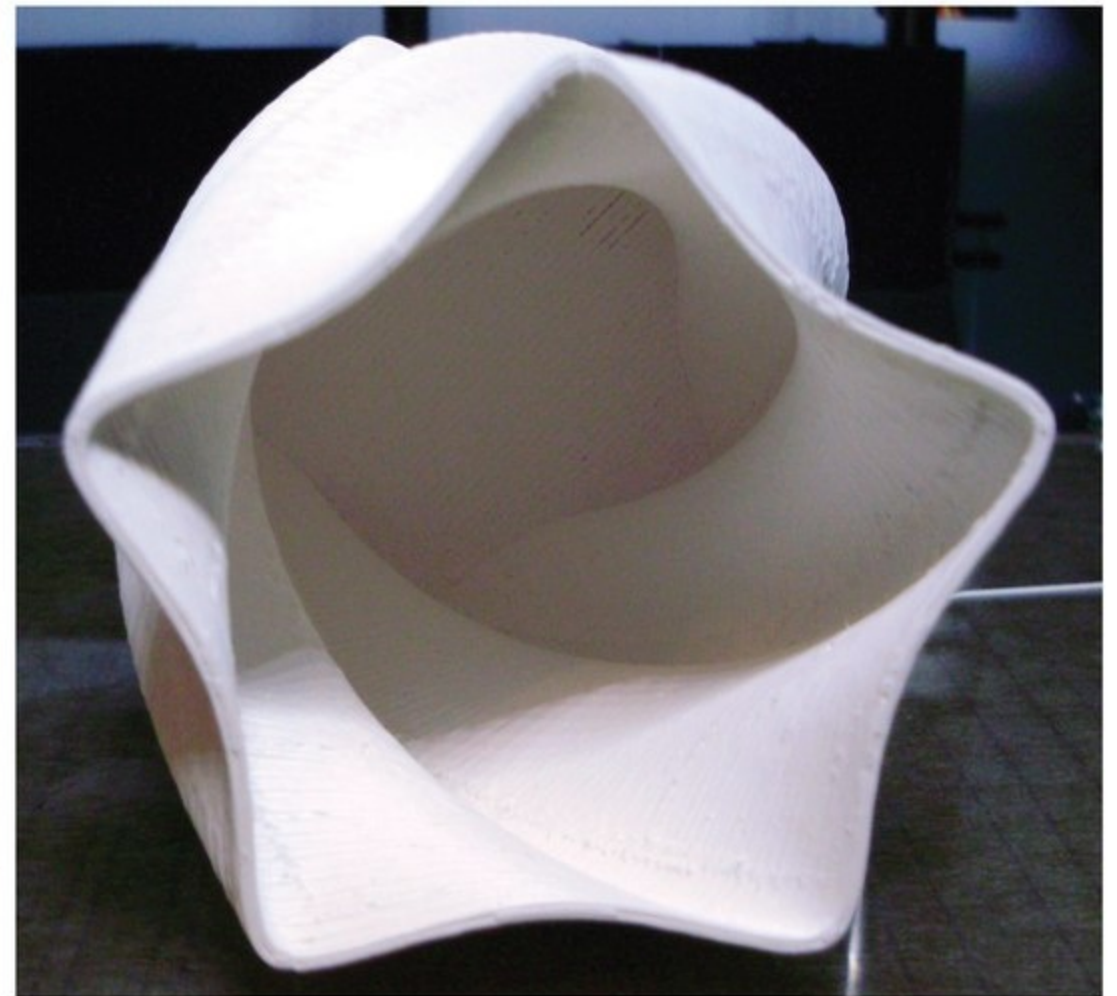
Maximum Build Volume (WxHxD): 7.8 x 7.8 x 7.8 inches
(20 x 20 x 20cm)
Printing Mode: Fine (100 microns), Standard (200 microns),
Speed (300 microns), Ultra Fast (400 microns)
Print Head: Single Nozzle
Nozzle Diameter: 0.4 mm
Print Speed: 150 mm/sec
Filament Diameter: 1.75 mm
Print Material: ABS, PLA
File Type: .STL and XYZ format
Panel Type: 4 x 16 LCM
Language: English and Japanese
Connectivity: USB 2.0 Wire
OS Support: Windows XP (.net 4.0 required) / Windows 7 or above
(for PC); Mac OSx 10.8 / 64-bit above (for Mac)
Hardware Requirements: X86 32/64-bit compatible PCs with
2GB+ DRAM (for PC); X86 64-bit compatible Macs with
2GB+ DRAM (for Mac)

da Vinci printer website / XYZ Printing, Inc.
http://us.xyzprinting.com/us_en/Home

TinkerCad — Easy to use online CAD Software from AutoDesk
<https://tinkercad.com>

Thingiverse
www.thingiverse.com

This is a very useful site that is full of links to 3D printable files.
www.hongkiat.com/blog/download-free-stl-3d-models



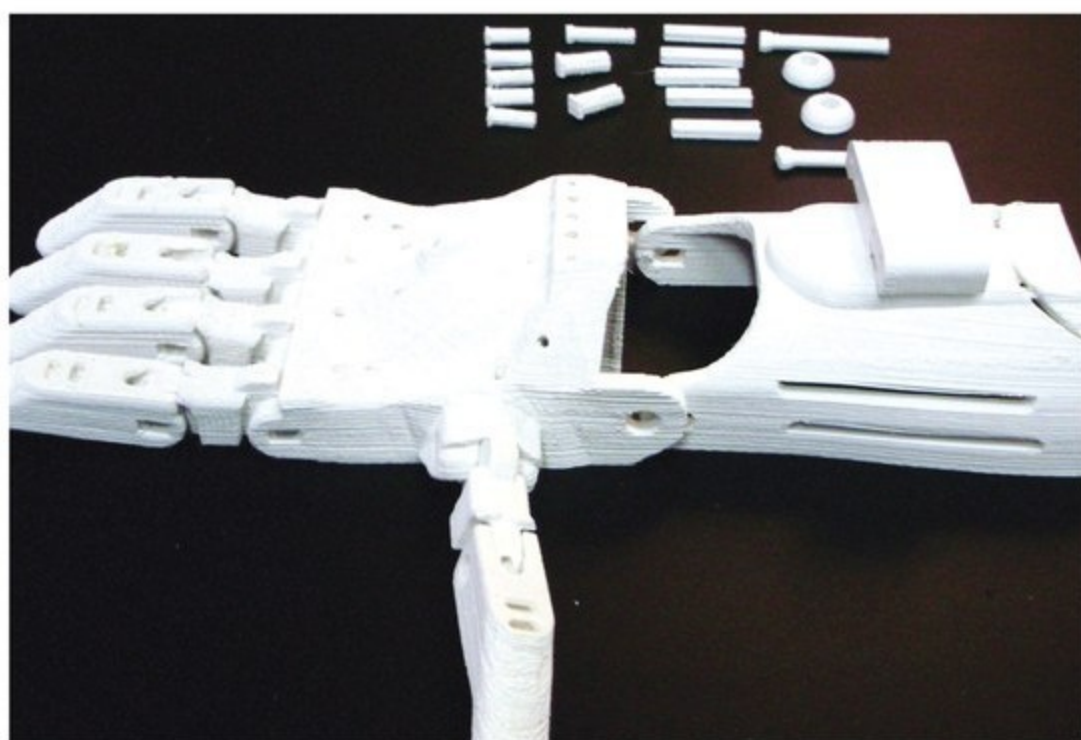
Different views of the star vase demo print.

Of course, during this process, I was also searching the Internet for CAD software that was easy to understand. I did download and try playing with a couple of them and found that I have a lot to learn. (My previous use of CAD involved Cardboard Aided Design — basically, cut the shape out of cardboard, lay it on a sheet of steel, trace around it with some soapstone, then cut the plate with an acetylene torch or a handheld plasma cutter, and then grind it to clean it up/fit.)

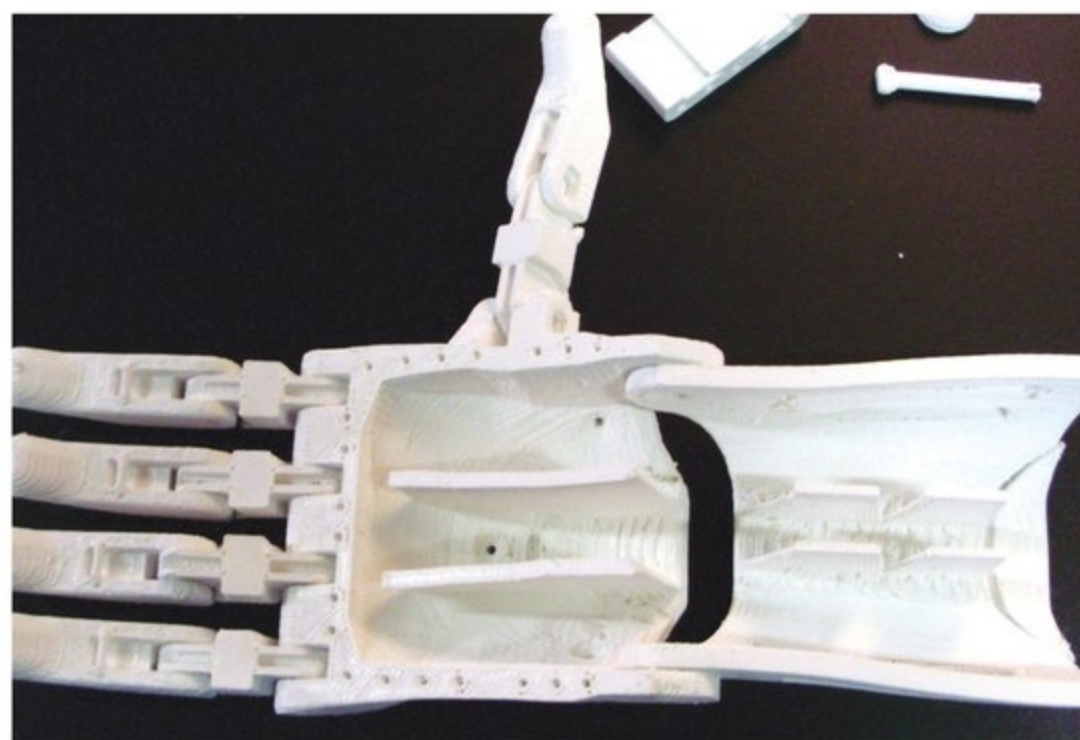
I then discovered an Internet-based CAD program called TinkerCad provided by the kind folks at AutoDesk. So far, it is the one I have been playing with since it was made for children — it's actually pretty easy to use. Another thing that I learned is there are many websites that have the STL files that are used by 3D printers.

STL files can be downloaded and then unzipped and imported into the da Vinci's XYZ slicer software and prepared for printing. I've downloaded several objects from a website called **Thingiverse.com** that is provided by MakerBot. Over the next couple of days, I explored over 25 websites that have STL files and CAD files. There are literally thousands of things online that you can download, modify, and print.

Eventually, I decided to *really* test my new 3D printer. I printed an entire eNable human prosthesis/robotic hand called the Raptor that has 30 plastic parts. I learned a lot from that printing job and had a few failures. The failures



The eNable Raptor hand (top).



The eNable Raptor hand (bottom).

taught me about changing some of the settings for print quality and resolution in the da Vinci 1.0 XYZ Slicer software. Eventually, I assembled the entire hand and got it working, and am now in the process of hacking it for robotics use, of course.

The part of all this that still amazes me is it was all

made possible by sharing knowledge with a fellow builder over the years, and the unbelievably generous gift Jerome Miles sent me out of the blue. The Internet can really bring people from all over the world together, and open source knowledge can "enable" folks to do things they never thought was possible. It's pretty cool when the student becomes the teacher ... **SV**

JOIN TEAM SYNERGY MOON!

Synergy Space Explorers are the power that drives the Synergy Moon space program. We are Team Synergy Moon, building a mission to the moon that includes Micro Satellites, Lunar Rovers and a Lunar Lander. We're going into space this year, and we're going to the moon with our Google Lunar XPRIZE mission.



You will have the opportunity to participate in space research, exploration and development missions, which currently include our Google Lunar XPRIZE mission to the moon, the Artemis NanoSat Constellation project and our remote controlled Tesla Orbital Space Telescope.



Get on board now and be part of this great adventure!

DIY SATELLITES AND SPACECRAFT SYSTEMS
info@synergymoon.com

CELEBRATING *a* DECADE *of* MAKING

10th

ANNUAL

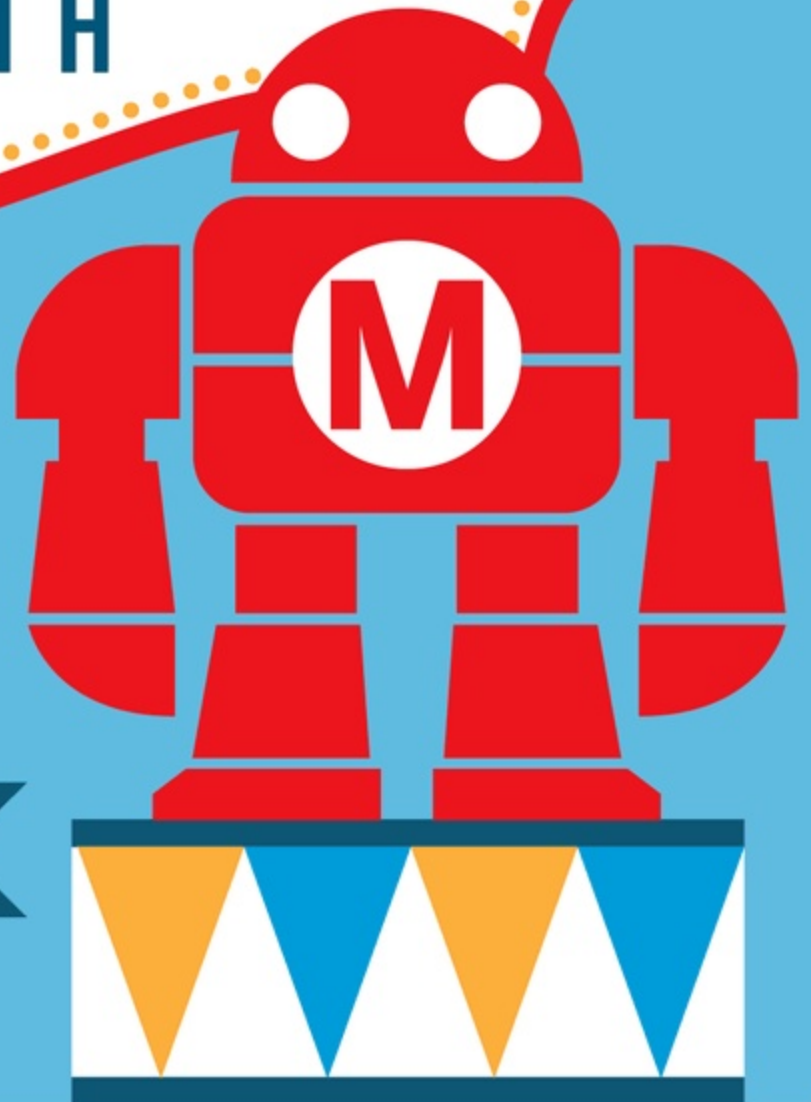
BAY
AREA

GREATEST
SHOW & TELL
ON EARTH

MAY 16+17

SATURDAY 10AM-8PM
SUNDAY 10AM-6PM

makerfaire.com



Brought to you by
Make: magazine

Maker Faire®

PRESENTED BY 

GOLDSMITH
SPONSORS

 AUTODESK

 iROBOT®
DRIVE THE FUTURE

ITA®
ITALIAN TRADE AGENCY

SILVERMITH
SPONSORS

 NVIDIA

ROBLOX



CGTrader's 3D Robot Challenge

The challenge required creating robot 3D models and uploading them to **CGTrader.com**. Participants competed for the Best 3D Robot model and Best 3D Robot portfolio. In the latter category, designers with more models we are more likely to become the winners. Each category had a winner and two runner-ups.

The challenge ran from December 22, 2014 to February 15, 2015. Designers competing had the chance to win these prizes: Motion Design Bundles by Video Copilot; iClone Animation Pipeline Toolkits by Reallusion; two KeyShot5 licenses by Luxion; one CGSchool video from V-Ray or 3ds Max by CG School; 200 credits; and promotional features by CGTrader. The runner-ups also got rewarded with prizes such as 3D Shader Bundles by Video Copilot; iClone5 PRO Bundles by Reallusion; Video of choice from V-Ray or 3ds Max classes by CG School; and 100 credits by CGTrader.

"Over the past years, robot models have become the fourth most popular category on CGTrader.com and we

believe it is an amazing topic to capture the imagination of 3D designers who are fascinated by both machines and video games. We see the 3D Robot Challenge as a great opportunity to look into the future, improve your skills, and use downtime to play and create for the futuristic theme," commented Marius Kalytis, CEO of CGTrader.

CGTrader has been constantly encouraging 3D designers to participate in various challenges to foster creativity and imagination. Its challenges have helped designers discover new themes, as well as win amazing prizes, and have even produced several world-firsts.

Currently, CGTrader also runs the 3D Systems Art Challenge, 3D Architecture Challenge, and 3D Printed Kids' Toys Challenge, where designers can submit both paid and free 3D models.

Check out their website for current and future events.

SV



Khiew Jit Chun
www.khiewjitchun.wix.com/jckhiew3d

As artificial intelligence and robotics increasingly take the spotlight in near-future predictions, CGTrader launched a 3D Robot Challenge for its community of over 120,000 3D designers in order to design and visualize potential robots of the future. The challenge encouraged designers to explore the sci-fi theme with the most advanced technological creatures that could well be our neighbors, counterparts, or even enemies in the coming decades.



Brumak
www.cgtrader.com/Lordlight



Titan
www.cgtrader.com/crashmgn



Spider robot
www.cgtrader.com/3d-models/industrial/machine/spider-robot-2



Khiew Jit Chun
www.khiewjitchun.wix.com/jckhiew3d

ROBOGAMES 2015!



www.robotgames.net

2015

RoboGames

Returns!

The International RoboGames has been an annual event held in the Bay area in California since 2004. This is the largest robotic event in the world (Guinness World Records says so!) and has enjoyed support from the viewing public, along with builders and teams throughout the globe. The event includes many aspects of robotics all under one roof, which allowed for learning and interaction with other disciplines. If you built a line following robot, you could talk with the Sumo builders. If you were interested in the hockey or soccer robots, you can spend time looking over the combat robots. There literally was something for everyone, and many people (fans and builders alike) looked forward to this event year after year. So, you can imagine the sadness within the robotics community when this event did not return for 2014!

A large event of this type requires a lot of logistical support, hours of preparation, an attention to detail on a huge scale, and, unfortunately, a lot of money. Sadly, for 2014 these things were in short supply, and the RoboGames event did not happen. However, the good news is that RoboGames is returning for 2015!

The international event will be held April 3rd through April 5th (Friday, Saturday, Sunday) at the San Mateo, CA Fairgrounds. All three days

will be action packed, with all the different classes and competitions you have come to expect.

While the combat contests tend to generate the most interest, there are a lot of varied events that have something that will appeal to just about any builder. If you are under 18, there are multiple events that you can enter for free! To get a listing and an approximate schedule go to <http://robogames.net/events.php>.

Something new for 2015 and that made this comeback possible was a Kickstarter campaign that was

successfully funded for the purpose of filming the event. This will be used to create a Web series, and for DVD and Blu-Ray sales. This video coverage will focus on the heavyweight combat class, and this year's lineup is sure to be one of the best ever.

As of this writing, the heavyweight class is already full, with a great mix of veteran machines and some great new builds by established teams. The extra year off seems to have primed the community. The excitement over this event just keeps growing!

For example, will last year's heavyweight combat winner, Original Sin retain his title? The big hitters, Last Rites and Electric Boogaloo are returning — will there be another face-off? Will the inclusion of Robot Wars champion, Storm II throw a curve to the competition? Or, will any of the new bots built just for this event come through and take the gold?

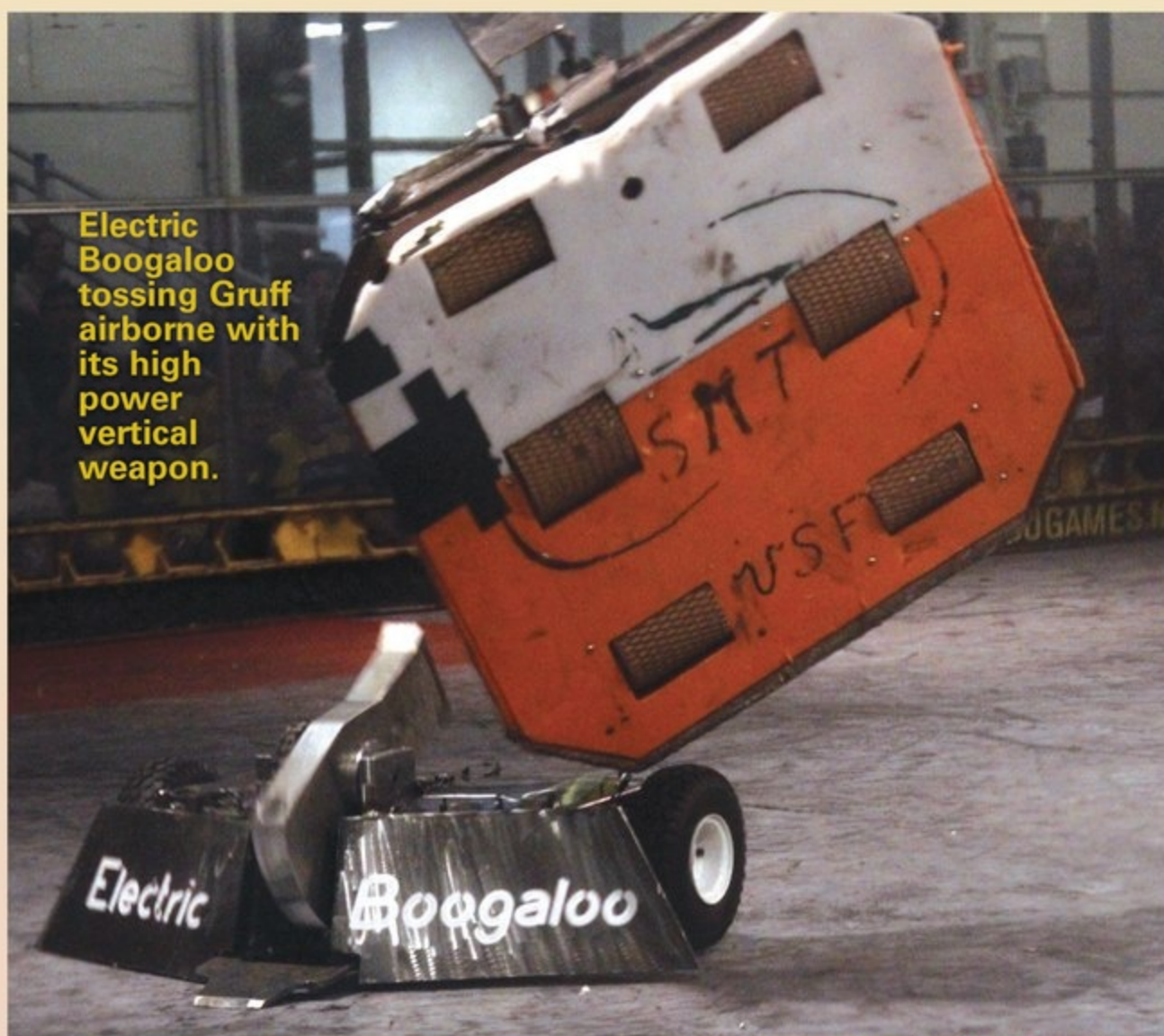
The field has never looked more competitive. This is an event not to be missed. Build a robot and compete, or come out to watch the action.

Get full event details at <http://robogames.net/index.php>.

SV



Raging Scotsman corners Dexter for some flame throwing action!



Electric Boogaloo tossing Gruff airborne with its high power vertical weapon.



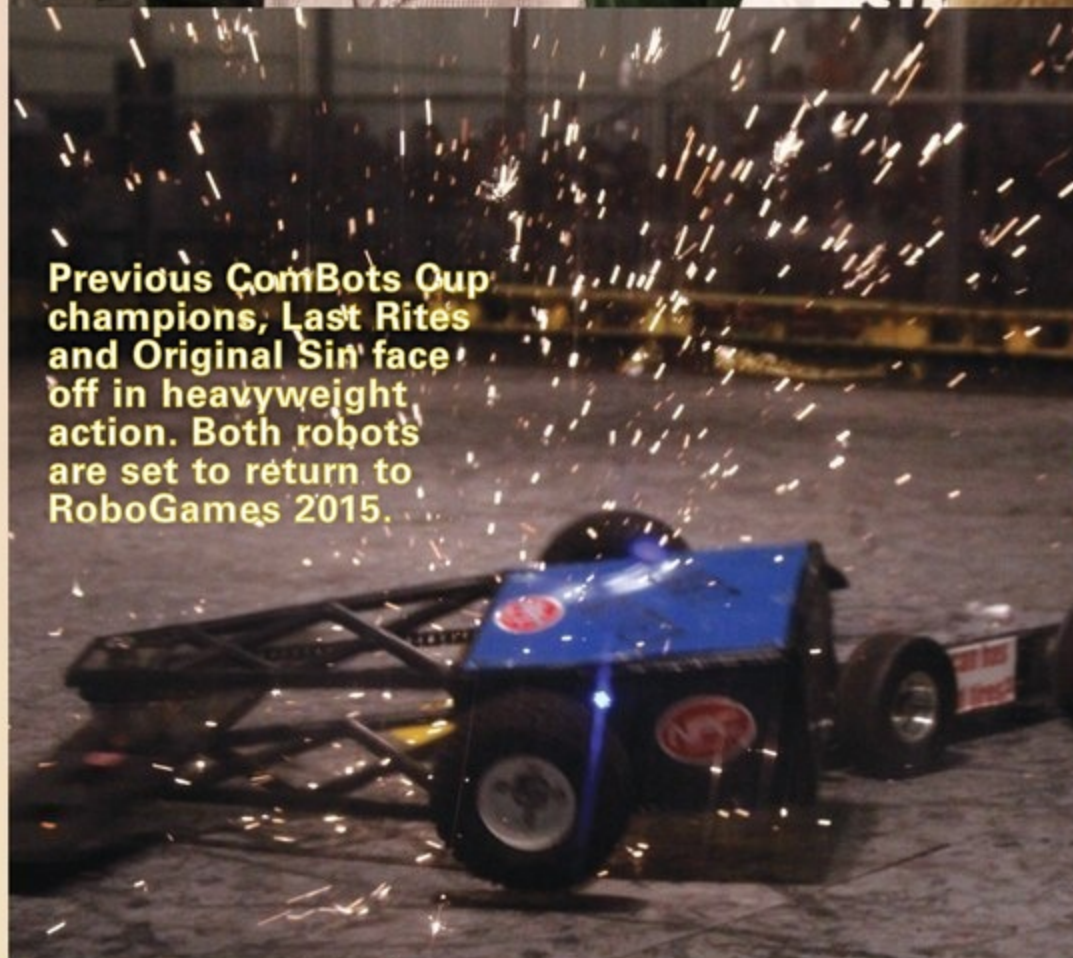
In addition to all of the combat action, there are multiple classes featuring less destructive pursuits. The humanoid robots are always entertaining.



Crowds enjoying the action in the combat arena. For some of the more intense matches, it can be standing room only!



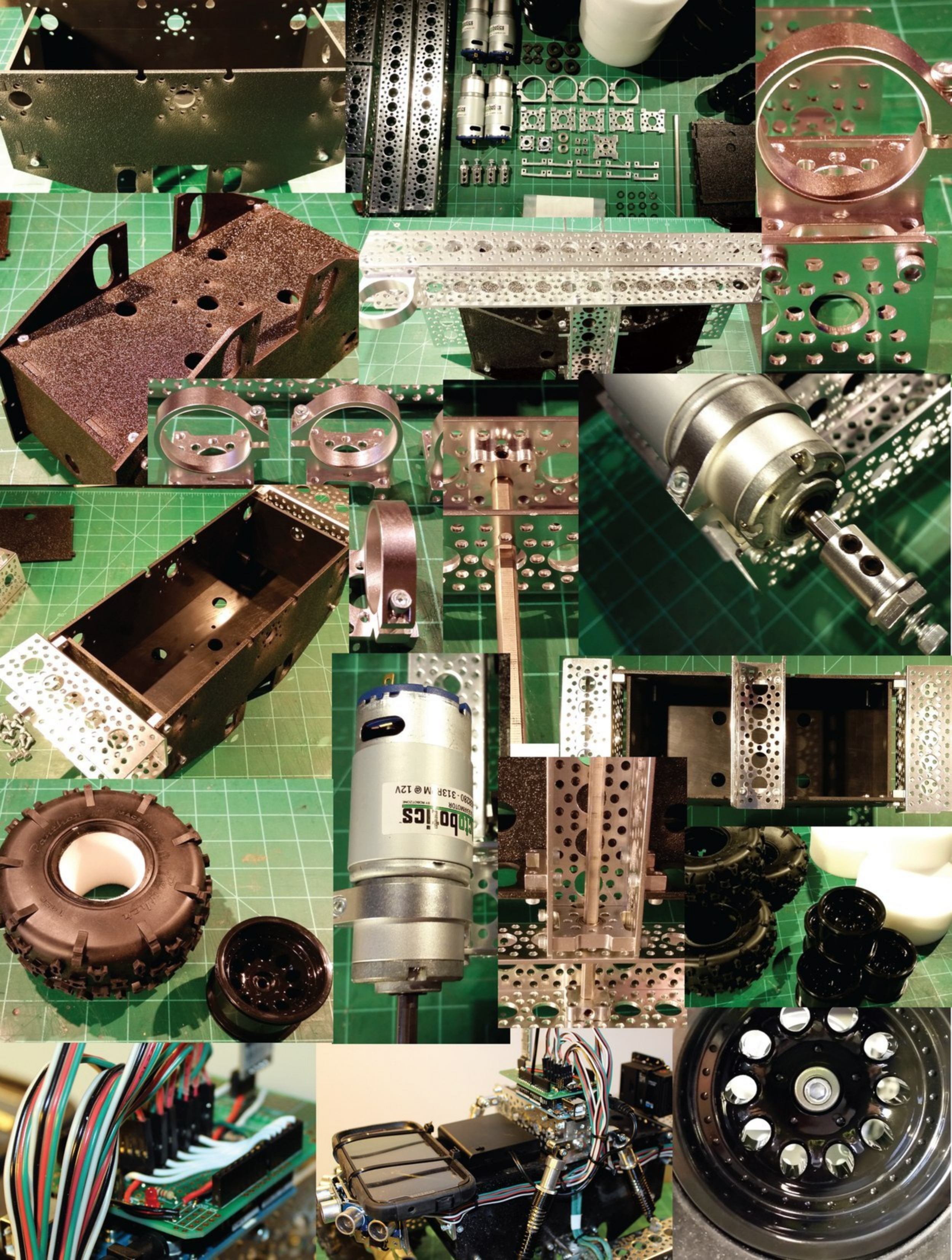
Flame effects weapons are always a crowd favorite!



Previous ComBots Cup champions, Last Rites and Original Sin face off in heavyweight action. Both robots are set to return to RoboGames 2015.



Here, Gruff gives former champion, Sewer Snake a ride with its lifting arm. Gruff will be returning to RoboGames 2015, but Sewer Snake's builder, Matt Maxham has decided to build an all new bot for this event.



REVIEW

A Capable Chassis for Your Next Project

By Jeff Cicolani

Post comments on this article at
www.servomagazine.com/index.php/magazine/article/march2015_Cicolani.



I have been looking at off-road robot chassis for a while. So, when the opportunity to try out the new Nomad chassis from Servo City was presented to me, I jumped on it. Nomad is a four-wheeled frame built for rugged outdoor use, constructed with parts from Servo City's Actobotics line of channel and hardware. Actobotics parts are something else I've been wanting to try.

Building the Nomad

The package containing the Nomad arrived shortly after Thanksgiving, along with a number of Black Friday purchases from some of my favorite robotics suppliers. This was going to be a fun robot-filled holiday!

Going through the parts, one thing that was missing was any form of printed instructions. As a robotics hobbyist, you get used to the fact that most of the items you order will not come with instructions. If you're lucky,

you may get a pin diagram with your microcontroller, but even that is less and less common. Most companies are using the Internet to provide their documentation. So, naturally, I turned to the Servo City website (www.ServoCity.com).

I did not find the Adobe PDF with the instructions as I was expecting. Rather, all that was available was an assembly video. This is not my preferred format for assembly instructions, but it would do. I proceeded to clear



Figure 1. The Nomad kit is packaged in many bags.

my work surface so I could spread out the parts and make room for my tablet that would be used for viewing the assembly video.

As I began building, the first thing that really struck me was the quality of the parts. The channel had very precise holes in a pattern that allows for a lot of options. The mounting hardware and brackets were exquisitely milled from aluminum. All in all, the precision, strength, and quality of the Actobotics system is impressive.

The second thing that struck me was the pile of plastic bags that were now covering my bench (**Figure 1**). They were everywhere. It took me a while to remove the parts from all of the bags. There were a lot of bags. Many of them were far too large for the parts inside.

With all of the parts removed, it was time to get down to the fun part: building this beast. I busted out the hex keys and started to get to work. It was at this point I discovered I did not have the correct key. The 6-32 hex head screws require a 7/64 hex key which is not one of the sizes in any of my sets.

Fortunately, I discovered one of my torx head bits

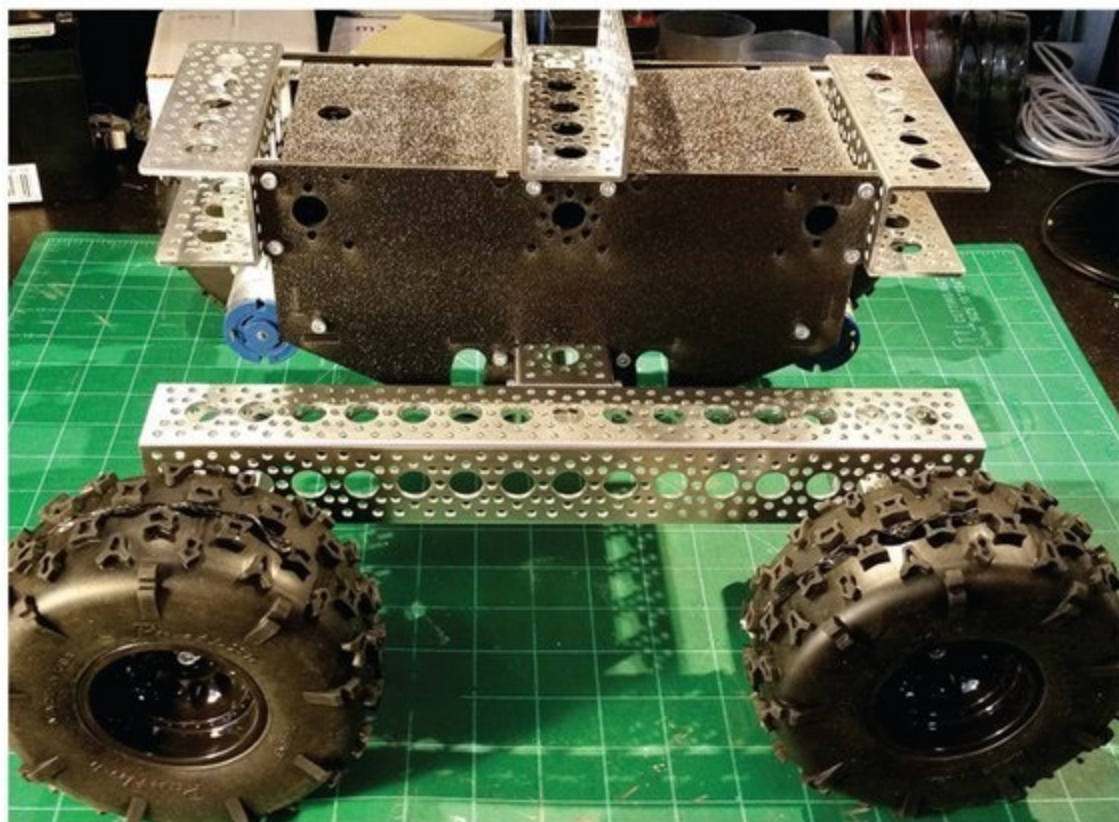


Figure 2. The completed Nomad chassis.

worked nicely. Undaunted, I forged ahead with the build.

Assembly was straightforward. The ABS plastic box which makes up the body went together easily. The brackets aligned nicely with the holes in the channel. The provided bearings snapped easily into place. All in all, the precision with which everything went together was brilliant.

However, shortly before assembly was completed, I ran out of screws. At first, I thought I must have tossed out a bag of screws with the other bags (there were a lot of bags remember). Checking the packing slip in the box, I discovered they sent three bags of 25 screws; a total of 75. To put the kit together as demonstrated in the video requires 88 screws. I was 13 short. A quick call to Servo City straightened the whole thing out. They sent me another package of screws and updated the kit to make sure all future shipments contain the right number.

Screws in hand, I was back to work. When I was finished, the assembled kit was larger than I was expecting. Putting it together you are aware of the size, but when you get the large knobby wheels onto the four 12 volt geared motors, somehow it feels bigger.

At this point, I had a very impressive chassis (**Figure 2**). The kit is just that: a chassis. It comes with the frame, box, four wheels, and four motors. Any electronics needed to get it moving are entirely up to you to provide. For this review, I wanted to keep it simple since the idea was to test the Nomad chassis as it comes in the kit.

To get it going, I simply dropped in a 12V sealed lead acid (SLA) battery, an R/C remote receiver, and a dual channel motor controller. For safety and convenience, I threw in a toggle switch, as well.

In addition to the Nomad kit, Servo City also provided me with a RoboClaw dual channel 15 amp motor controller by Ion Motion Control. This, too, is a new addition to their extensive product line. I have to say I was as excited about the RoboClaw as I was about the Nomad.

This was an impressive controller with a number of modes including R/C, direct, and serial. The controller was well documented on their website (www.ionmc.com), so getting it connected and configured was very easy. Once the RoboClaw was set up, it was time to take the Nomad out for a test spin.

Putting the Nomad Through Its Paces

I cannot express enough how much fun the Nomad is. The four motors are powerful and — for lack of a better term — downright zippy. The knobby off-road tires want to grip to everything, likely due to the soft rubber as well as the tread pattern. On more than one occasion, the Nomad tried to climb up my leg, a wall, or my neighbor's fence.

Due to my experience with other platforms that use a wheeled configuration with a tank drive format, I was expecting to have troubles turning. I could not have been

These are the additional parts I ordered for the suspension modification:

- 2 x 1/4 inch Bore Set Screw Hub (0.770 inch): Part # 545548
 - 2 x 1/4 inch x 4 inch D-Shaft: Part # 634080
 - 1 x .250 inch ID x .500 inch OD Flanged Ball Bearing (2 pack): Part # 535198
 - 1 x 1/4 inch Ball Bearing Quad Pillow Block: Part # 535130
 - 4 x 90 Degree Quad Hub Mount D: Part # 545324
 - 2 x 130 mm Aluminum Robot Shocks (pair): Part # 585032
 - 1 x Beam Attachment Block B (4 pack): Part # 585600
 - 1 x 2.31 inch Aluminum Beam (2 pack): Part # 585406
 - 2 x 45 Degree Dual Angle Channel Bracket: Part # 585426
 - 4 x 90 Degree Quad Hub Mount A: Part # 545420
 - 1 x .750 in L x 6-32 Zinc-Plated Alloy Steel Socket Head Cap Screw (25 pk): Part # 632120
 - 8 x 6-32 Machine Screw Nylock Nuts: Part # 90631A007
- You may also want to invest in a 7/64 hex key, part # 57185A44.

more wrong. It was very responsive and had no problems making sharp turns spinning in place or on carpet or grass.

I took it out to an empty lot near my home to run it through more paces. The Nomad seemed unstoppable. This thing was so much fun — right up until the point where one of the wheels fell off. Fortunately, this happened on the street and not in the field so I was able to find the screw that fell out. The screw that holds the wheel to the hub loosened and worked its way out. This was simply fixed by applying a little Loctite Blue to the screws. I noticed several of the other screws holding the motor mounts and a few other places were coming loose, so I put some Loctite on them as well.

Once everything was back together, I had to take it outside and run it around some more. I took it down to another lot where construction was about to begin so I could test it on more challenging terrain: dirt mounds, construction debris, and other obstacles (and because it was so much fun). Again, the Nomad was unstoppable. Until a wheel fell off, again.

A couple of the set screws that hold the wheel hub to the motor shaft fell out. This time, however, it was in a mix of gravel and mud, so they were lost forever. Fortunately, Home Depot carries the 10-32 set screws, so I picked some up and got Nomad back on the road. I ran it all around the neighborhood and finally down to a boat dock. Overall, it went nearly a mile before the charge on the battery dropped low enough to cause it to struggle with hills.

Next-Level Nomad

With all the initial fun out of the way, it was time to start examining the Nomad for a practical application. First, I wanted to tackle one of the things that I did not like about this particular design. The way the Nomad's

suspension works is quite simple. The box that serves as the body is mounted to a piece of channel that then attaches to two more aluminum channels that run perpendicular. The motors are mounted at the ends of these channels and the wheels mount directly to the output shaft of the gearing.

One of these channels is mounted on a pillow block with a pair of bearings, providing smooth pivoting. This allows that side of the Nomad to adjust to its environment and provide excellent traction. However, the other channel is rigid to the frame. This means the body is always pitched to align with that side of the chassis. The design is effective, but I wanted more.

To add a little more flexibility to the platform, I ordered a few more brackets, two pairs of small shock absorbers, and the hardware necessary to make both channels pivot and a few other parts for mounting the sensors (see the **sidebar**).

The shock absorbers are another new addition to the Actobotics line. The shocks limit the movement of either channel, but the sum of the angles with these in place exceeds that of the single pivoting design. Plus, it looks cooler (**Figure 3**). With the suspension modified to my liking, the next step was to make the robot conversion.

My next goal is to build an autonomous robot that will live up to the Nomad name. This final project will consist of a microprocessor with a selection of sensors to allow Nomad to traverse various types of terrain. It will be able to navigate autonomously in one of two modes: Magellan and Roam.

In Magellan mode, Nomad will navigate from one point to another via GPS, with the ability to stop at waypoints along the path. In Roaming mode, it will explore its surroundings, mapping obstacles and terrain. In addition, it will be equipped with a grasper to allow it to interact with

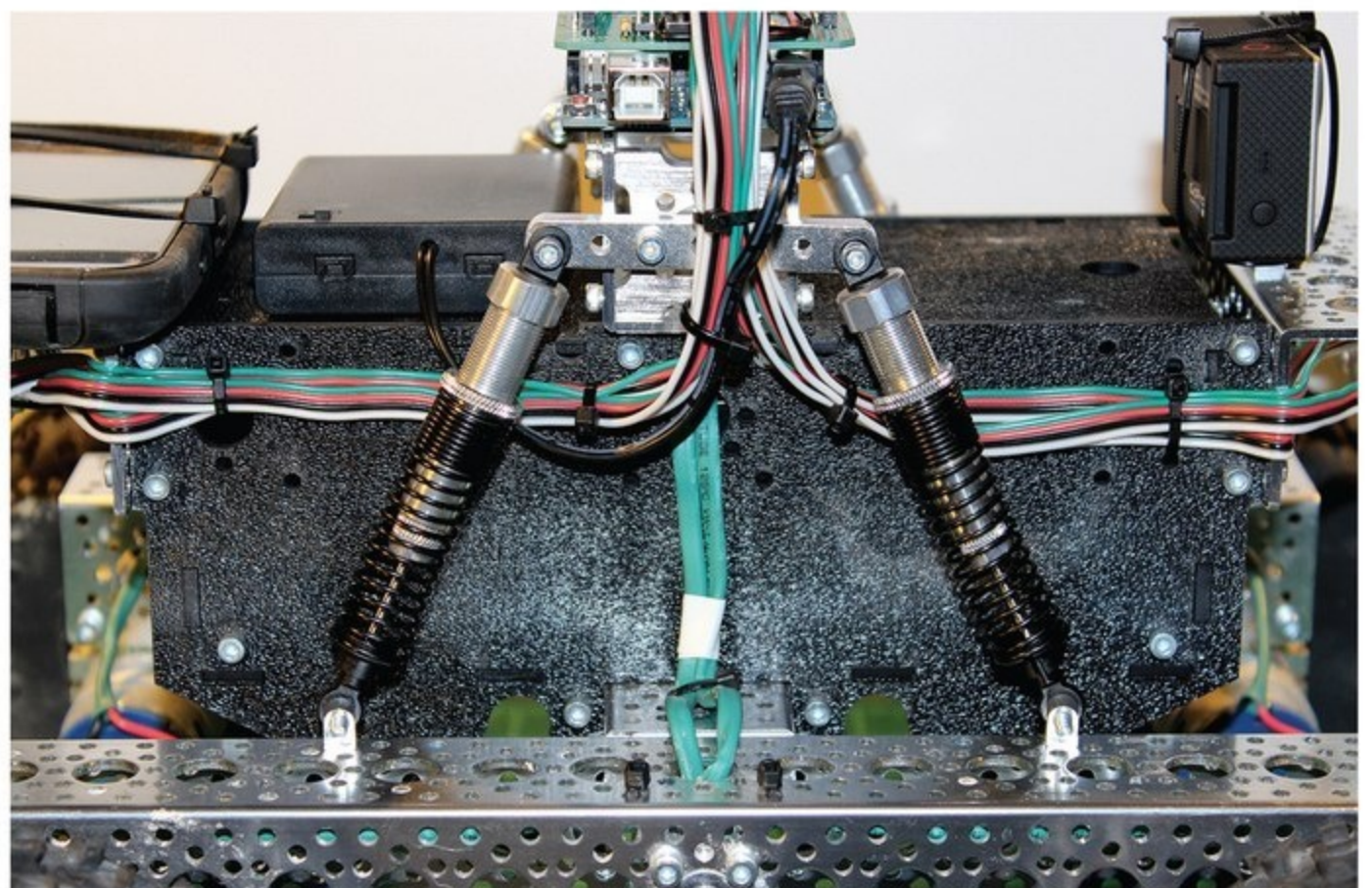


Figure 3. Modified suspension to add a little flare.

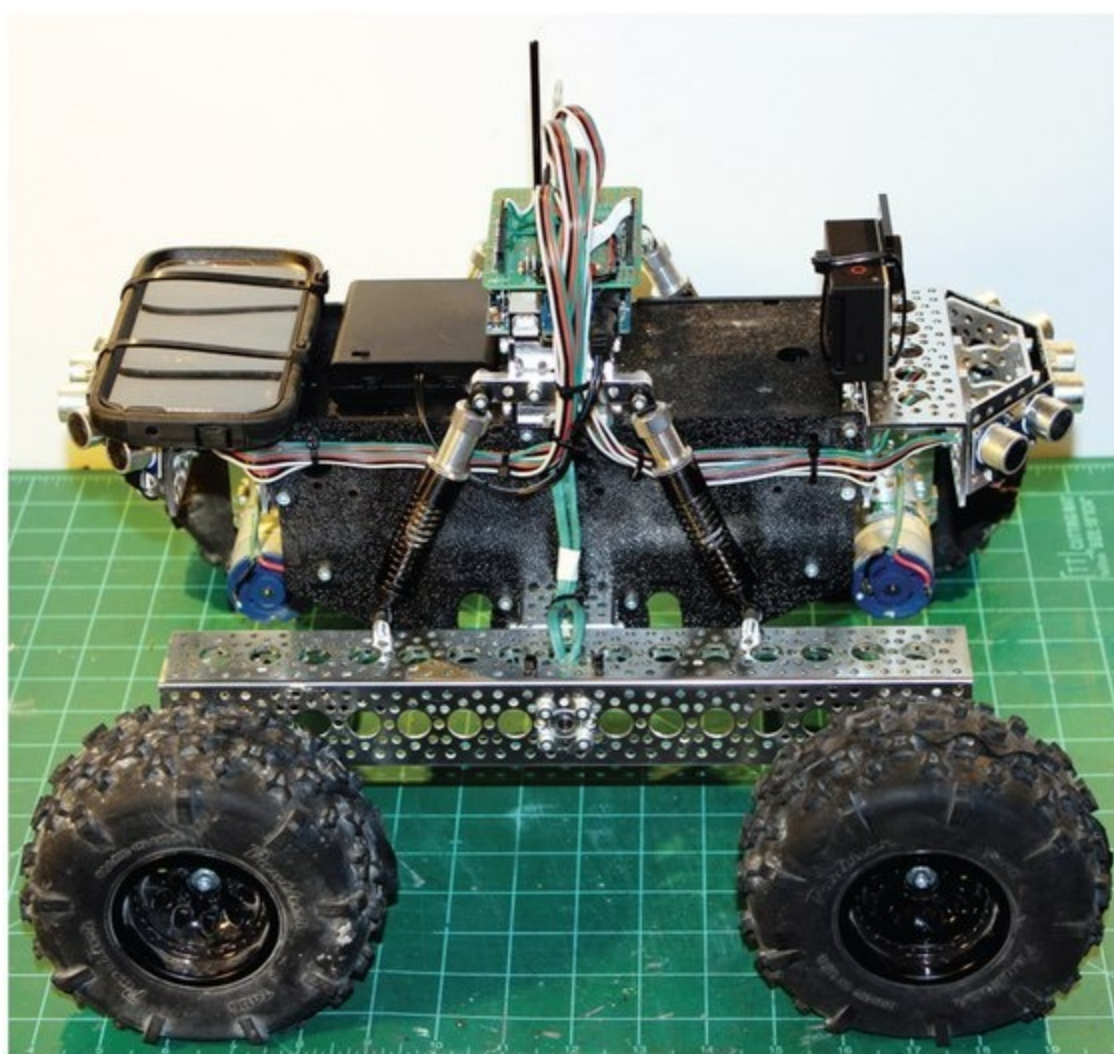


Figure 4. The completed Nomad robot conversion.

the environment. These are some challenging goals to be sure, but Nomad has the flexibility to handle this and more.

When I upgraded my mobile phone recently, I kept the old one. This will become the primary processor on the robot. With GPS, accelerometers, gyroscopes, and other sensors built in, it will make the perfect platform on which to build a robot. I mounted an Arduino and add a Seeed Studio proto shield. The proto shield is configured with the connections needed for the six sonic range finders — three in front and three in back — as well as the Bluetooth transceiver used to communicate with the phone.

With the large SLA battery, motor controller, 15A



Figure 5. This tire kept separating from the wheel.

Servo City
www.servocity.com/index.html

Nomad Assembly Video
www.youtube.com/watch?v=FAPDkyeAek8

Nomad Product Page
www.servocity.com/html/nomad_4wd_off-road_chassis.html#VK4mTCvF-FM

toggle switch, and R/C receiver in the body, there was no room for more electronics. The Arduino and shield were attached to the channel mounted on top of the box. The phone was zip tied to the back of the box. Just for fun, I also mounted a GoPro camera to the front. The completed robot is shown in **Figure 4**.

With everything mounted, I took it down to the park to test the new configuration. It worked beautifully and was able to conquer any terrain I put it up against. I will need to make a cover for the electronics and a better mount for the phone, however.

I did have an issue with the rubber of the tire wanting to separate from the wheel (**Figure 5**). The lip on the wheel is not quite deep enough to prevent the tire from breaking bead and popping off. This generally only happened when I made sharp turns or spun in place. The tires can be super-glued to the rims, which is a simple fix to this separation issue.

My "final" remedy for this will actually be to add a couple of plates on either side of the tire, attached by screws that pass through holes in the wheel. This will, in essence, create a bead lock which should prevent the tire from popping off of the wheel. That being said, all in all I am very happy with the end result.

Summary

The new Nomad kit from Servo City is a very capable platform for use in most any wheeled robot application. If you are going to build one, be sure to invest in a tube of Loctite Blue. You will also need to provide a solution for the tire separating from the wheel. These are minor issues, however.

The Actobotics line of parts from which Nomad is built allows for limitless modifications to the kit. If you're not looking to build a Magellan bot as I now am, the kit can be customized to virtually any application. As mentioned, you do have to provide your own electronics and power source, but that is common for most chassis in this class.

I am impressed with the quality of the Actobotics parts I used from the kit and the added ones used to customize it. I will definitely be looking to Servo City and Actobotics for parts for future projects. **SV**



Imagine it. Build it. Control it.™ Fly it!

Introducing the Quadrino Nano UAV Flight Controller



Introductory
price of
\$99 USD *

- ATmega 2560 (256Kb flash @ 16MHz)
- Compact, six layer PCB design
- Integrated vibration damping
- MultiWii and Arduino compatible for DIY projects
- InvenSense MPU9051 (Gyroscope, Accelerometer, Magnetometer)
- MS5611 Barometer with foam cover
- Venus 838FLPx 50Hz GPS



*Limited time offer. Price after promotion \$149.99 USD.

Available at www.Lynxmotion.com and www.RobotShop.com



Lynxmotion and DialFonZo are trademarks of RobotShop Distribution Inc.

Make a MiP Hack Back

By Dave Prochnow

When Mark W. Tilden left WowWee Robotics, pundits were quick to spell the death of commercial entertainment robots. This dire prediction was a little too quick it seems.

In late 2014, WowWee Group Limited had another commercial entertainment robot success story — MiP.





Officially not connected to an approved acronym, MiP is a two-wheeled, multifunctional,

autonomous, self-balancing robot. Think micro-sized Segway, but way cuter. Just plop four AAA batteries into its battery pack, flick the switch on, and set MiP down on its two wheels. Within seconds, it is balanced (by rocking back and forth on the two wheels) and ready for commands.

Forget the dog bone shaped remote control from Robosapien! MiP uses GestureSense commands for controlling its movements. Simply put, GestureSense means waving your hand in front of MiP: Move your hand away from the front sensors and MiP will drive forward; push your hand towards the sensors and MiP will move backwards; swipe your hand across the sensors (left or right) and the little two-wheeled robot will turn; finally, hold your hand rigidly in front and MiP will stop moving.

Don't fret! MiP isn't limited to GestureSense. It's also a robot with personality.

There are actually eight personality modes programmed in MiP: GestureSense (default mode), Tricks, Track, Cage, Roam, Stack, Sleep, and, lastly (in a beloved nod to Robosapien), there is a Dance mode.

Lacking a remote, how do you access these modes? Ingeniously, dual encoders on the MiP wheels enable mode selection by rotating a wheel clockwise or counterclockwise. The selected mode is indicated by an RGB LED embedded behind an opaque chest plate on MiP. For example, Dance mode is indicated with a light blue or turquoise color.

If the Dance mode isn't enough to satisfy your Robosapien flashback, MiP also "speaks" in a primitive language. Not quite the prehistoric grunts of Robosapien, but incoherent enough to earn the respect of seasoned WowWee robot owners.

Finally, MiP is able to connect to your smartphone or tablet via a

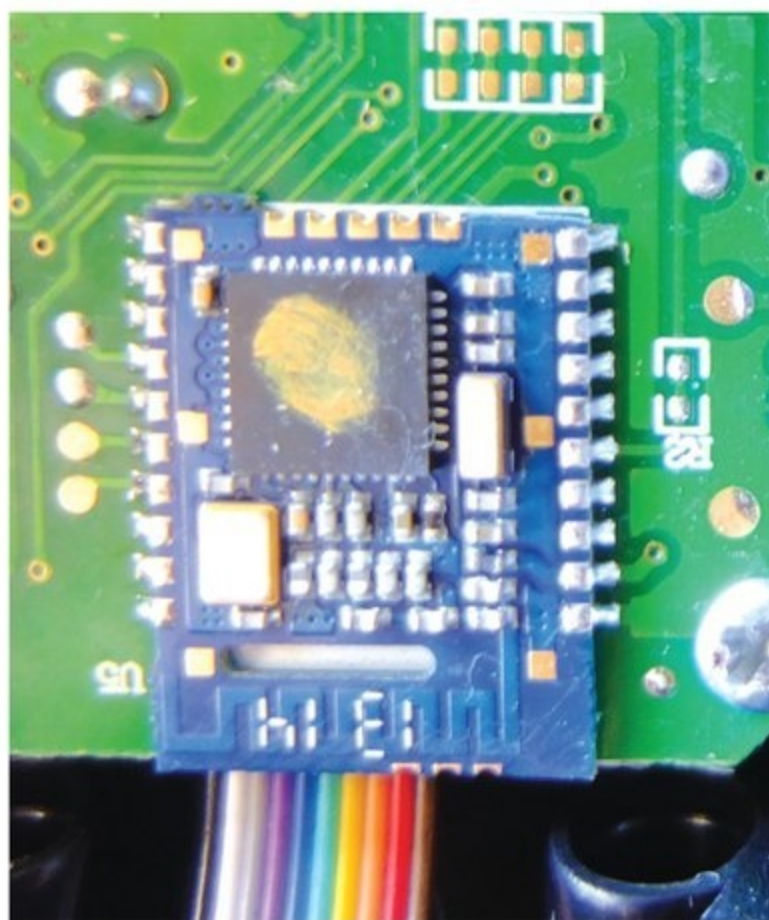


Figure 1. UART blue — a hacker's paradise hidden inside MiP.

couple of apps. Available as either an iOS (version 7.x and higher) or Android App (version 4.3 with Bluetooth Low Energy and higher), the MiP app adds an incredible amount of value to the MiP experience. Sporting additional behaviors and games, the app is also able to monitor the robot's battery current — an invaluable asset when operating MiP for extended periods of time.

Sadly, Windows phone and Blackberry owners don't have a native app for controlling MiP. Additionally, Android owners should be aware that not *all* 4.3+ devices will work with the MiP app.

For example, low cost tablets like the Apex TM772 will *not* work. Even though this particular tablet has

Bluetooth, it is lacking the Low Energy Protocol (BLE).

Therefore, be sure to check the Android compatibility list on the official MiP website located at www.wowwee.com/mip/compatibility before downloading the app.

MiP Back Hack

Thankfully, the flattering nods to Robosapien don't stop with a couple of robot firmware features. MiP also fully embraces the hacking spirit imbued by Tilden in the original Robosapien.

Secreted away on the main printed circuit board (PCB) is a small, four-pin, blue Universal Asynchronous Receiver/Transmitter (UART) connector. Gaining access to this UART is fairly easy and well within the ability of *SERVO Magazine* readers. **Figure 1** shows the MiP UART connector.

Complete instructions for opening MiP and gaining access to the UART connector are described/illustrated on the SparkFun Electronics website (click on the Tutorials tab link). There are two different tutorials: one for the SparkFun MiP Proto-Back, and another for the SparkFun MiP Proto-Pack.

Although very similar in appearance, these tutorials are very different. The "Pack" version reuses the MiP battery pack and is very limited in its practicality. Conversely, the "Back" tutorial is a goldmine for all wanna-be MiP hackers.

There are three very powerful modifications that you can easily make

Parts List

SparkFun MiP Proto-Back (DEV-13066)*
MiP Expansion Cable — 4" (PRT-13149)*
MiP Expansion Socket — 4-pin JST (PRT-13150)*
USB Battery Pack — 1,000 mAh (PRT-11358)*
Four-Pin Right Angle Male Header (PRT-00553; this is a 40-pin break away header)*
One Scrap USB Cable
(4) M3x45 Socket Head Bolts
(2) Black Cable Ties — 4"

*Can be purchased from SparkFun Electronics — sparkfun.com

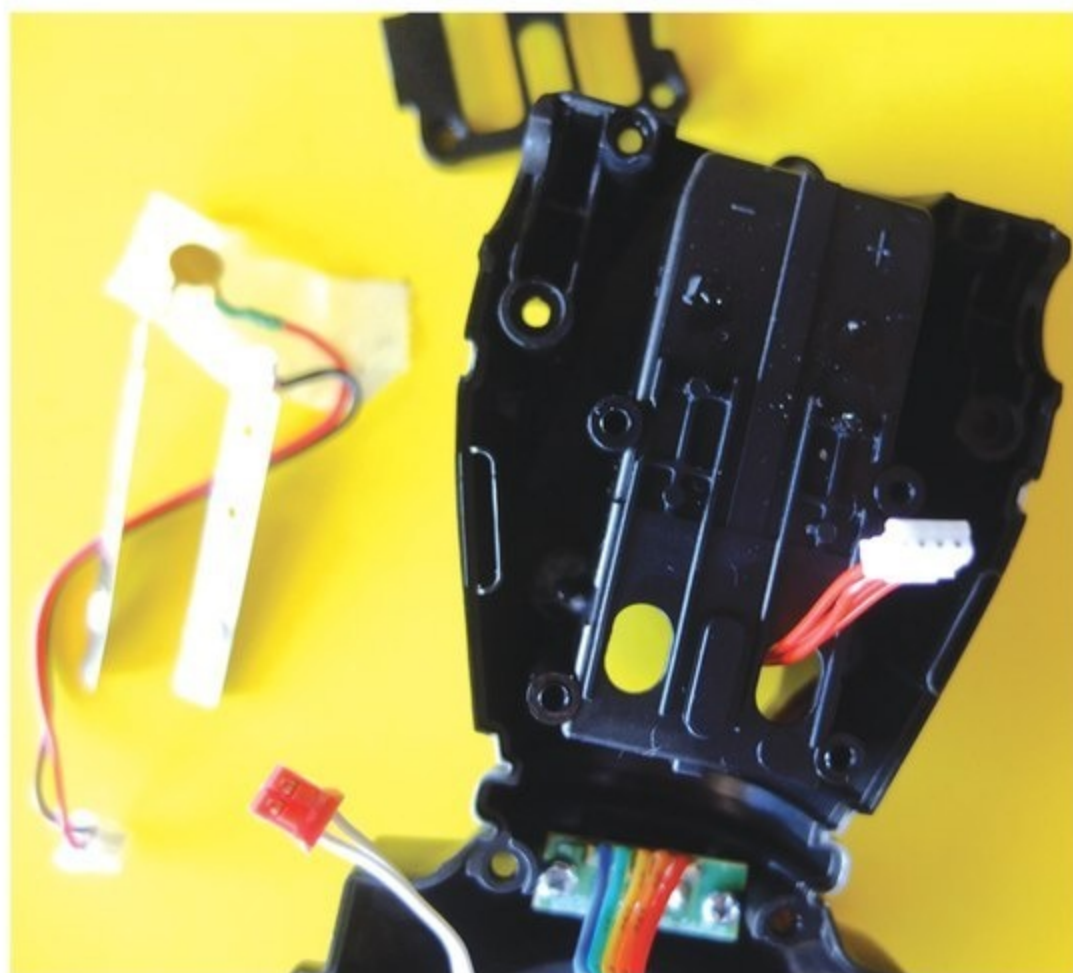


Figure 2. Remove the battery pack terminal strips for routing the expansion cable.

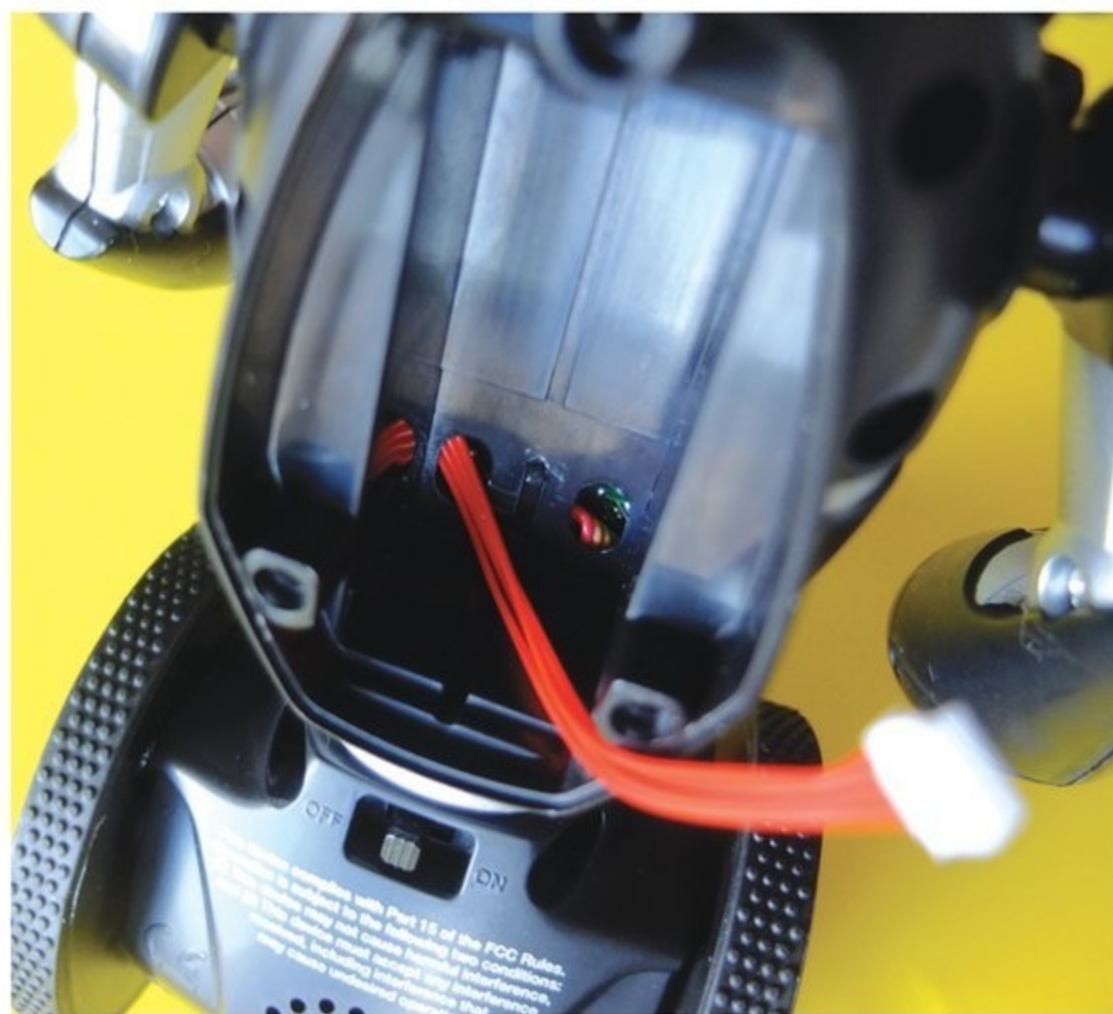


Figure 3. The MiP back is now ready for attaching the Proto-Back.

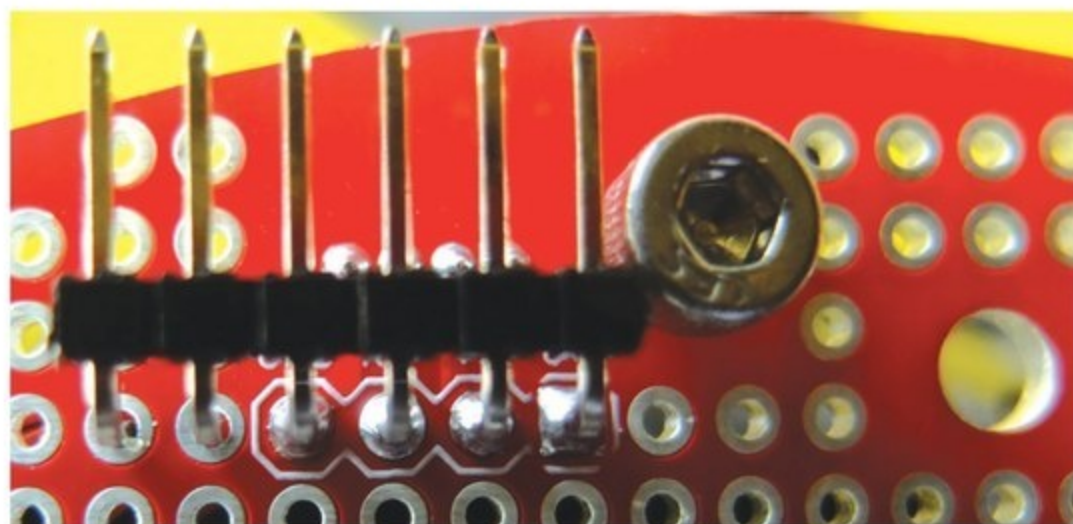


Figure 4. Solder a right angle-header to the expansion socket's pads on the Proto-Back. I used a six-pin header to allow me a couple of extra connections for a future project.

to the Proto-Back tutorial. These mods will give you an effortless connection to the MiP UART, install a raised PCB for prototyping projects on MiP, and — most importantly — install a rechargeable battery pack on MiP.

Modification 1. When it comes to routing the MiP expansion cable outside the main PCB area, don't cut any robot plastic. Instead, remove the battery pack positive/negative terminals and poke the cable through either one of the two generously sized holes that remain.

The terminals have been removed in **Figure 2**, while **Figure 3** shows the expansion cable routed through the MiP battery compartment.

Modification 2. Speaking of external battery packs, add a four-pin right angle male header to the Proto-Back on the MiP expansion socket pads for soldering the power pins (e.g., black/negative/GND and

This sweet little modification will enable you to return MiP to the factory-built battery pack at a future time (if needed). Furthermore, this mod allows you to power MiP via an external battery pack.

red/positive/6V) from a scrap USB plug. I used a six-pin header in **Figure 4**; either will work. This plug can then be used with a rechargeable USB battery pack.

While this unit outputs 5.5V at 700 mAh, it is capable of driving MiP. Plus, it's rechargeable, so no more AAA battery replacements and disposal.

When the new battery pack is wired up, it can be attached to the Proto-Back board with two cable ties. The attached battery pack is shown in **Figure 5**.

Modification 3. Once the MiP expansion cable has been installed through the battery compartment, it's

MiP Developer links

WowWee Group Limited has created a special "developer's site" for MiP hackers to explore. These Github repositories are vital resources for gaining access to MiP via Bluetooth or UART.

WowWee MiP Bluetooth Low Energy Protocol
<https://github.com/WowWeeLabs/MiP-BLE-Protocol>

MiP iOS SDK
<https://github.com/WowWeeLabs/MiP-iOS-SDK>

MiP BLE Protocol Command Table (very valuable stuff)
<https://github.com/WowWeeLabs/MiP-BLE-Protocol/blob/master/MiP-Protocol.md>



Figure 5. Connect a scrap USB cable's power wires (i.e., black and red) to the proper Proto-Back pads (i.e., GND and 6V).

time to reassemble the robot. Replace only four of the original MiP back screws (i.e., the lower four back screws in the wheel housing).

Use the remaining four holes for mounting the Proto-Back to the MiP back. I used four M3x45 socket head bolts for attaching the Proto-Back as shown in **Figure 6**.

Lacking access to these fasteners, you might consider using 4-40 hardware with thin diameter standoffs as an alternative. Either way, make sure that you connect the expansion cable to the expansion socket prior to attaching the Proto-Back to MiP.

It would probably be a good idea to solder the four-pin header in place *before* you attach the back, too. Although (with a steady hand), you can solder the header following attachment of the back.

When these three modifications are complete, you will have a rechargeable MiP that is able to support additional hardware via a serial connection. The modded MiP in **Figure 7** is ready for balancing more

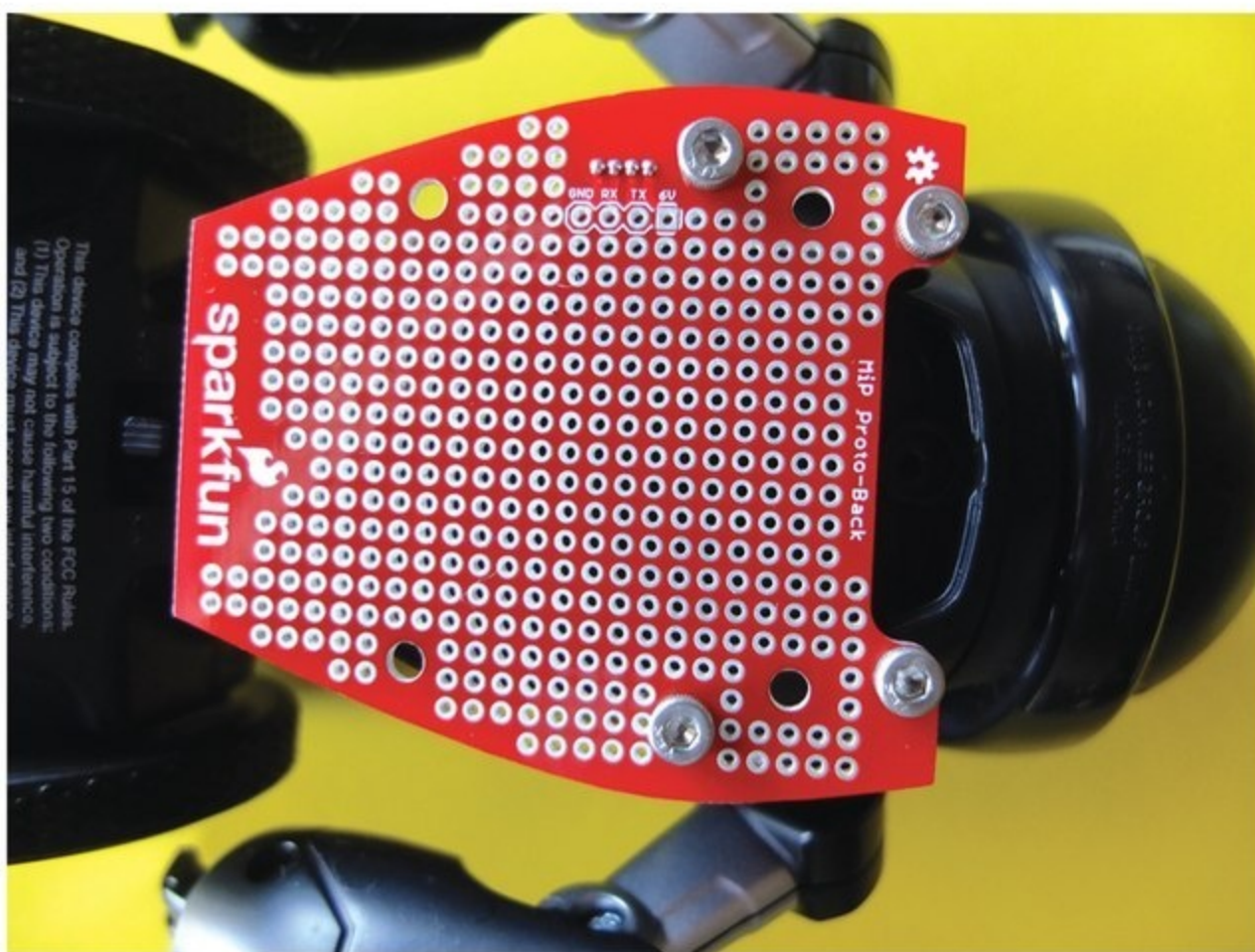


Figure 6. Attach the Proto-Back with four M3x45 bolts driven into the top four retaining screws on MiP.

hardware on its back.

If you elect to go this connected microcontroller route, your software first needs to send the byte 0xFF, then send/receive commands following the BLE protocol. That's it!

While not as tall as Robosapien nor as colorful (MiP is either black or

white), MiP is every bit as hackable. Once you've completed your MiP Hack Back mod, use the WowWee BLE protocol (see **sidebar**) for gaining access to MiP.

Your experimentation will help ensure that future generations of commercial entertainment robots will be accessible to bot building enthusiasts. **SV**

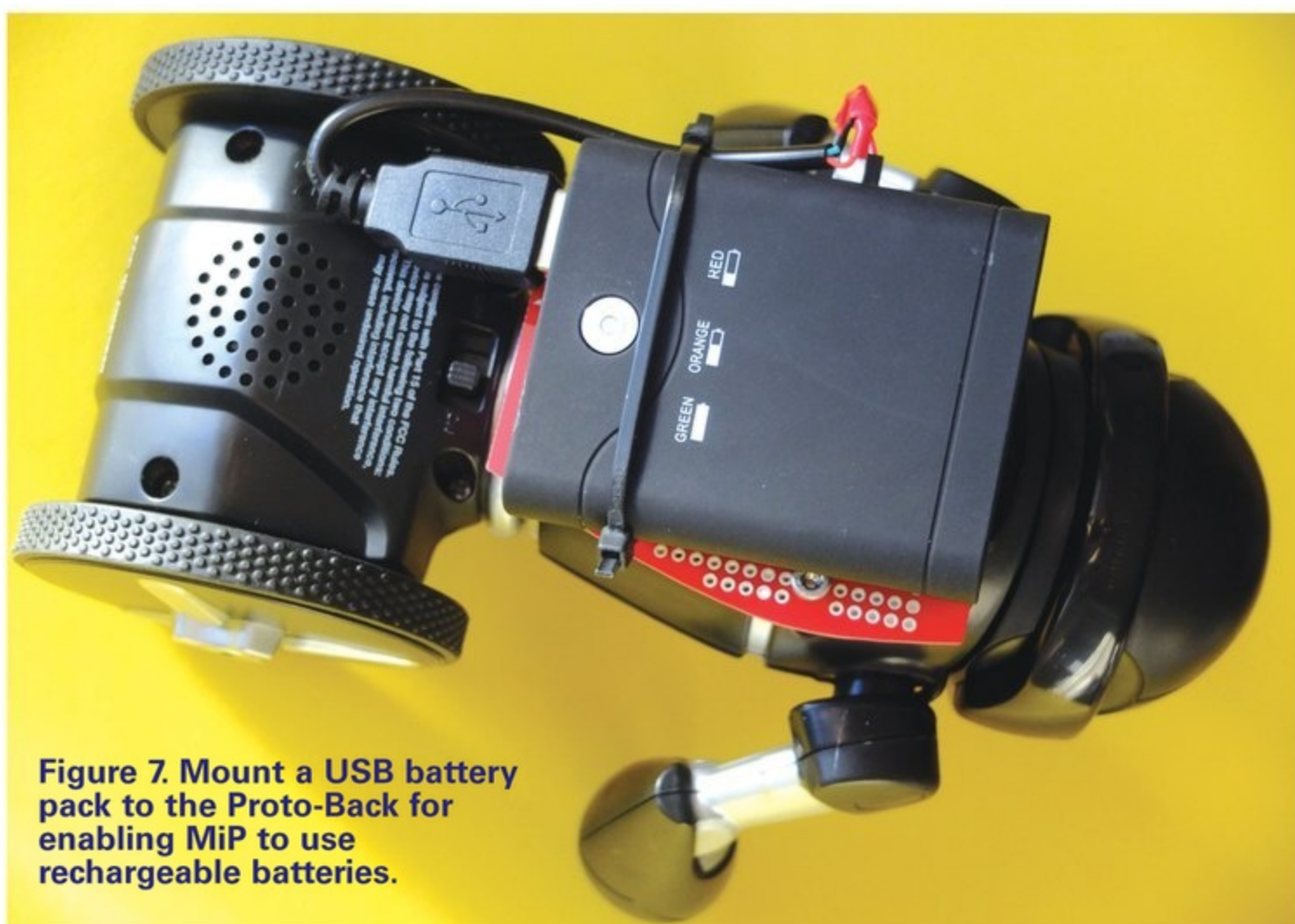


Figure 7. Mount a USB battery pack to the Proto-Back for enabling MiP to use rechargeable batteries.

The *SERVO* Webstore

CD-ROM SPECIALS

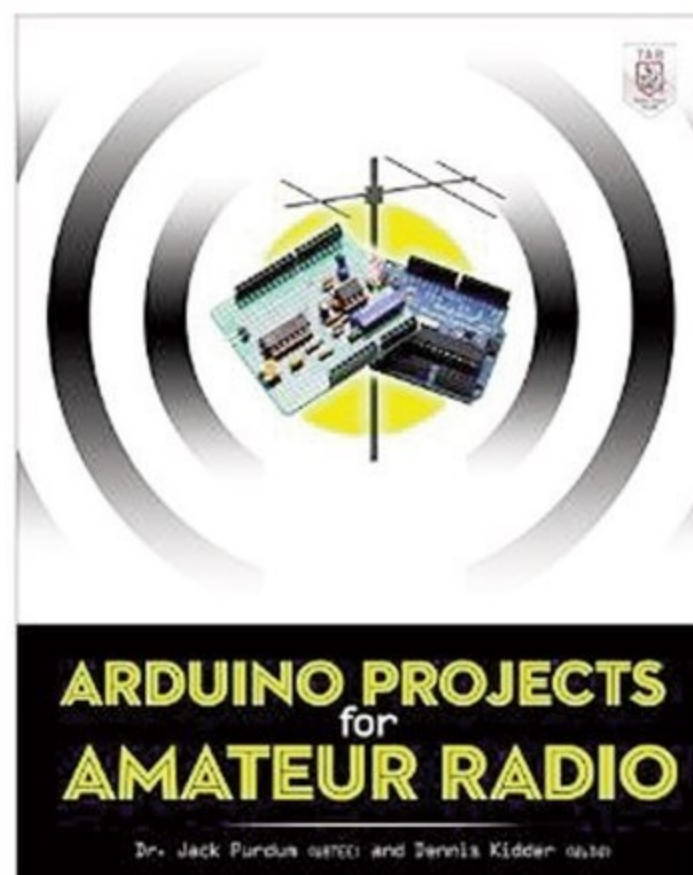
**11 CD-ROMs
&
Hat Special**



Free Shipping!

**Only \$ 229.95
or \$24.95 each.**

3.2015 FEATURED



**ARDUINO PROJECTS
for
AMATEUR RADIO**

Dr. Jack Purdum (W7ECI) and Dennis Kiden (W4JN)

ROBOTICS

Arduino Robot Bonanza

by Gordon McComb

Create high-tech walking, talking, and thinking robots. "McComb hasn't missed a beat. It's an absolute winner!"

Breathe life into the robots of your dreams — without advanced electronics or programming skills. *Arduino Robot Bonanza* shows you how to build autonomous robots using ordinary tools and common parts. Learn how to wire things up, program your robot's brain, and add your own unique flair.

\$29.95

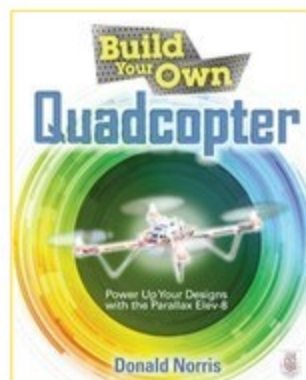
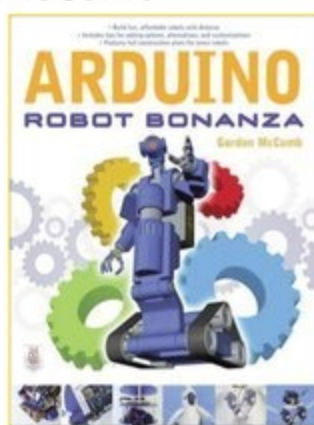
Build Your Own Quadcopter: Power Up Your Designs with the Parallax Elev-8

by Donald Norris

Build a custom multirotor aircraft!

Build and customize radio-controlled quadcopters that take off, land, hover, and soar. *Build Your Own Quadcopter: Power Up Your Designs with the Parallax Elev-8* features step-by-step assembly plans and experiments that will have you launching fully functioning quadcopters in no time.

\$50.00

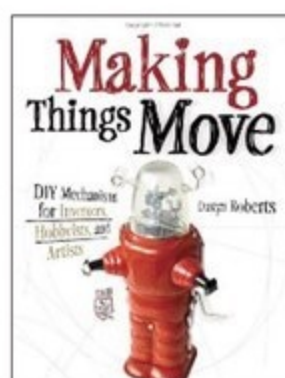


Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists

by Dustyn Roberts

In *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, you'll learn how to successfully build moving mechanisms through non-technical explanations, examples, and do-it-yourself projects — from kinetic art installations to creative toys to energy-harvesting devices. Photographs, illustrations, screenshots, and images of 3D models are included for each project.

\$29.95



Build Your Own Humanoid Robots

by Karl Williams

GREAT 'DROIDS, INDEED!

This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than \$300. Together, they form the essential ingredients for making your own humanoid robot.

\$24.95



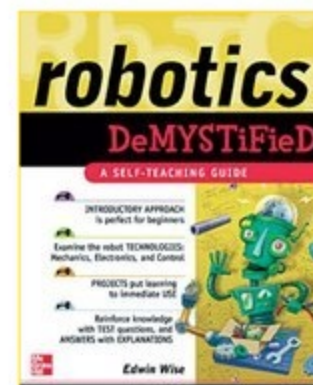
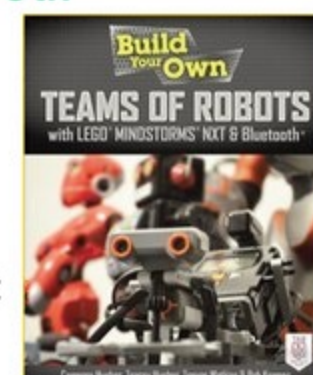
Build Your Own Teams of Robots with LEGO® MINDSTORMS® NXT and Bluetooth®

Plan, design, assemble, and program robot squads that communicate and cooperate with each other to accomplish together what they can't do individually. Inside the authors show you how to construct a team capability matrix (TCM) and use the Bluetooth Robotic-Oriented Network (BRON) so your robot teams can share sensors, actuators, end effectors, motor power, and programs. **\$30.00**

Robotics Demystified

by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS! Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **\$19.95**



We accept VISA, MC, AMEX,
and DISCOVER
Prices do not include shipping and
may be subject to change.

Or order online
<http://store.servomagazine.com/home.php>

SPECIAL OFFERS

Arduino Classroom Arduino 101 Projects Kit

Only
\$44.95



From Smiley's Workshop

Forbidden LEGO

by Ulrik Pilegaard / Mike Dooley

Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse.

Reg \$24.95 Sale Price \$19.95



Beginners Guide Book Combo.



Only \$85.95

Plus
FREE Priority Mail Shipping
US Only

"THE PHANTOM DRAW"

The KILL A WATT meter is the best way to help you determine your actual energy draw in ON and OFF home appliances.



Only \$29.95

To order call

1 800 783-4624 or online www.servomagazine.com

PROJECTS

The SERVO Buddy Kit



An inexpensive circuit you can build to control a servo without a microcontroller.



For more information, please check out the May 2008 issue or go to the **SERVO** webstore.

Includes an article reprint.
\$39.55

3D LED Cube Kit

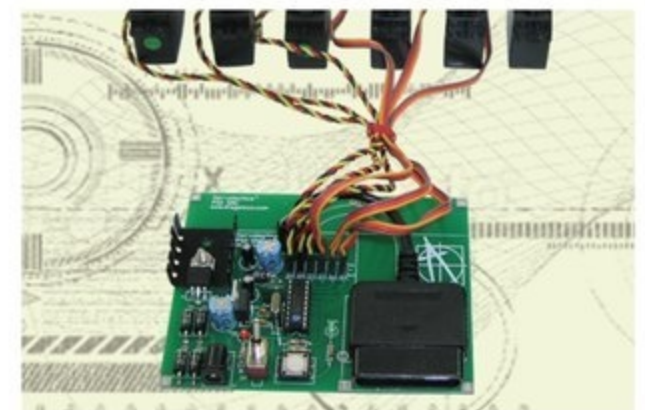
From the article "Build the 3D LED Matrix Cube" as seen in the August 2011 issue of *Nuts & Volts Magazine*.



This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes. Colors available: Green, Red, Yellow & Blue. Jig and plastic cases also available.

\$57.95

PS2 Servomotor Controller Kit



This kit accompanied with your own PlayStation controller will allow you to control up to six servomotors.

Includes all components and instruction manual.

For more information, please see the February 2011 edition of *SERVO Magazine*. Assembled units available!

\$79.95

To order call 1-800-783-4624

SERVO Magazine Bundles



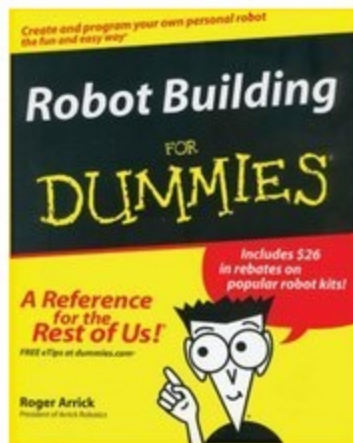
Save \$10.00
Only \$57.95!

Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, 08, 09, 10, 11 and 2012.

Robot Building for Dummies

by Roger Arrick / Nancy Stevenson

Discover what robots can do and how they work. Find out how to build your own robot and program it to perform tasks. Ready to enter the robot world? This book is your passport! It walks you through building your very own little metal assistant from a kit, dressing it up, giving it a brain, programming it to do things, even making it talk. Along the way, you'll gather some tidbits about robot history, enthusiasts groups, and more.

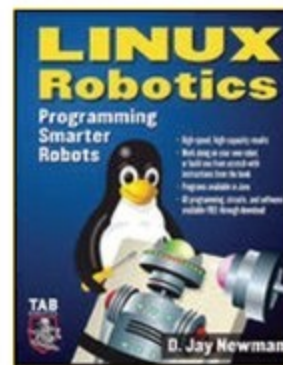


\$24.95

Linux Robotics

by D. Jay Newman

If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for “Zeppo,” a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need.

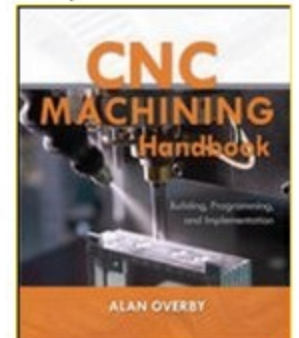


\$38.95

CNC Machining Handbook: Building, Programming, and Implementation

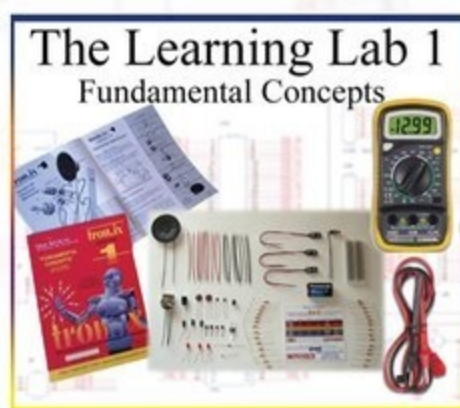
by Alan Overby

The *CNC Machining Handbook* describes the steps involved in building a CNC machine and successfully implementing it in a real world application. Helpful photos and illustrations are featured throughout. Whether you're a student, hobbyist, or business owner looking to move from a manual manufacturing process to the accuracy and repeatability of what CNC has to offer, you'll benefit from the in-depth information in this comprehensive resource.

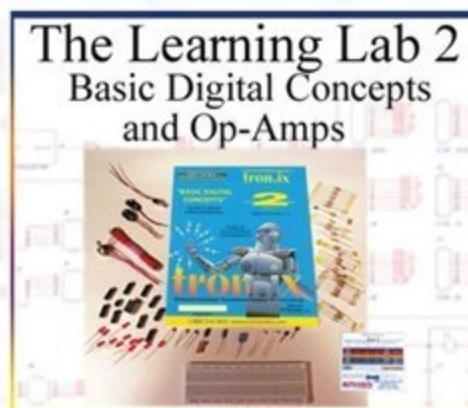


\$34.95

FOR BEGINNER BOT BUILDERS



\$59.95



\$49.95



\$39.95

The labs in this series — from GSS Tech Ed — show simple and interesting experiments and lessons, all done on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit, and continue to build your arsenal of knowledge with each successive experiment.

For more info and a promotional video, please visit our webstore.



DRAWING A LINE IN THE SAND

Beachbot — created by a team from Disney Research Zurich and ETH Zurich — autonomously creates sand drawings by using a retractable rake and a host of sensors.

Beachbot is only a prototype at this point, so it still needs a little assistance. You have to set



boundaries for Beachbot using four provided poles, then its built-in Wi-Fi and laser scanner tracks its position and keeps it within the lines. There are two rear wheels to get Beachbot going, while one front wheel is in charge of steering.

Beachbot — which is 60 cm in length and 40 cm in both width and height — can create designs within a 10 x 10 meter square area.

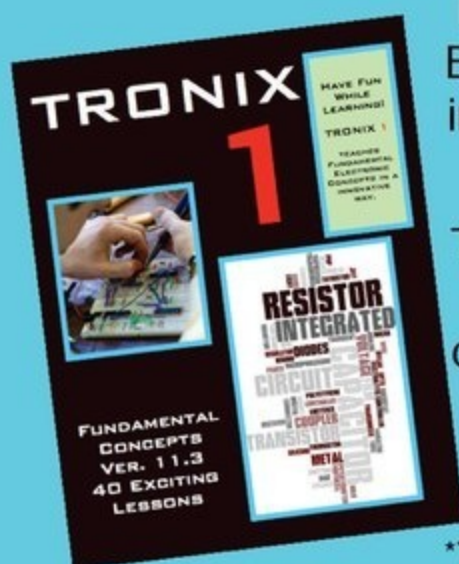
However, since its rake is retractable, Beachbot can create multiple lines — not just one continuous line. It can also create lines of varying thickness, from five centimeters to 40 centimeters in width.

If you look at the picture, you'll notice Beachbot doesn't leave any tracks behind it. It was one of the main challenges for researchers at Disney Research Zurich.

"Robot sand art is basically a path planning problem in robotics," explains Paul Beardsley, a principal research scientist at Disney Research Zurich.

Based on the lines of a drawing, the robot computes a trajectory that most closely approximates them. For big art pieces, the trajectory still has to be adjusted manually.

However, the group hopes to completely automate the process, so that you can give any picture to the robot, and it will generate a line drawing and compute the corresponding path.



Introducing the ALL NEW TRONIX 1 Lab. Bold and exciting full color illustrations that include interactive content specific to each lesson.

This exciting new curriculum is the perfect tool for your classroom, and with the full online interactive activities that bring each project alive right before your eyes.

SHIPPING NOW!!!

*TRONIX 1 is backwards compatible with Tron.ix 1 and Mr. Circuit curriculums

CHECK OUT OUR FULL COLOR KITS
Our entire line of Solder Kits have been completely revised and updated to include beautiful full color illustrations.
Order your **FREE SAMPLE KIT TODAY**
LFComponents.com/free

615-625-2885

LFComponents.com

LFComponents



Getting Started with Pascal on a PIC

This is surely the golden age of DIY electronics — especially for newcomers starting out on a shoestring budget. Free and open source software abounds for drafting schematics, designing printed circuit boards, simulating circuit operation, turning laptops into oscilloscopes, and more. Along these lines, not long ago, something caught my eye which seemed too good to be true: a full-featured Pascal compiler for the PIC and absolutely free of charge —no advertisements, no limitations, and no strings attached.

Called PIC Micro Pascal (PMP for short), this superior compiler is the creation of Philippe Paternotte who has made it readily available for download on the Web. What so boggled me once I paused to examine it with more care is that this is no stripped-down rendition of Pascal. It offers all manner of features found within the Turbo Pascal and Delphi languages for PCs. The code it produces is highly optimized throughout — generally better than what I could write from scratch in assembler. In short, if you've craved advanced structured programming on a PIC, yielding compact hex code results without having to reach for your wallet, this is the way to go!

I switched over to PMP from assembler and compiled Basic, and have never looked back. Here's why.

First, Pascal is a splendid language in its own right. It's logical, sensible, simple to learn, and makes a structured approach to firmware design painless. Even better — unlike some languages — Pascal is easy on the eyes; the source code is almost self-documenting.

The PMP implementation is incredibly complete. Consider just a few of its features:

- Follows standard Pascal closely.
- Modularity (procedures and functions), with local variables and versatile parameter passing.
- Rich set of data types, including floating point numbers.
- Data structures, such as arrays, records, even user-defined types.
- Tons of optimization features.
- An attractive editor with integrated

PIC datasheet lookups, chip pinouts, configuration menus, program trees, code folding, and other niceties.

- Libraries for handling LCDs, the ADC, serial communication, I²C and SPI, multiplexing, etc.
- Extensive set of compiler directives for automated custom compilation.

Unlike most free software, PMP comes with a very complete help manual — a couple hundred pages when printed out. It's exceedingly detailed and precise, but missing one key ingredient: a quick and gentle "what to do first" guide. Actually, PMP is extremely easy to use, but only after you've successfully written and Flashed your first chip.

Because this language and its editor do so much, when first firing it up your eyes are apt to glaze over and you'll wonder how to begin. Hence, this article. Join me as we explore how to get started. After that, the help manual in all its glory should be a snap to tackle.

Blinking an LED perhaps isn't the most glamorous way to begin, but it does have the advantage of keeping things so simple you can focus on the language and the editor exclusively. Let's shoot for that as our first example.

Get It and Install It

If you're into PIC microcontrollers, you probably already have the Microchip MPLAB on your computer. If not, download and install it (it's free, too) since PMP calls upon its internal assembler to generate the final hex code. By installing this first, PMP takes care of hooking things up for us. See the **sidebar** for details on where to get MPLAB. Observe that PMP will work with either the older version 8 or the more recent MPLAB-X version. I prefer the older program myself.

Then, download and install the PIC Micro Pascal software. It's all pretty painless, with no surprises along the way. When done, PMP and the MPLAB assembler should be properly connected, yielding a complete system that's ready to go.

Set Your Project Options

If you're itching to create your first PIC Pascal program, start up the editor now. From the menu bar at the top, choose **File...New...Project**. **Figure 1** shows what you'll see. In the usual Windows manner, create a subfolder for your new project. Then, in the box at the

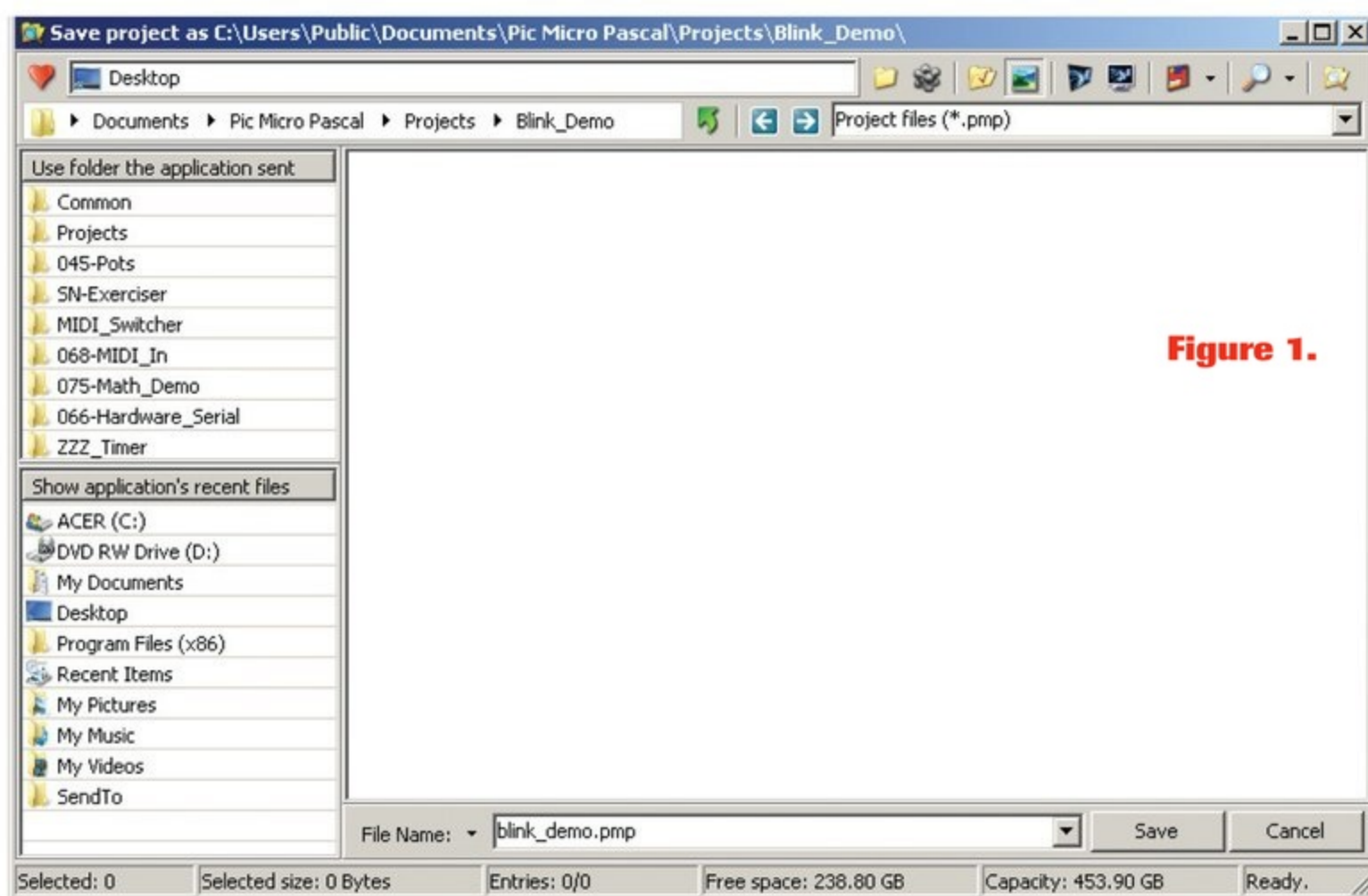


Figure 1.

bottom, enter a file name. Every PMP project has a project settings file (with an extension of .pmp) associated with it, containing all the configuration information. That's what you're creating here.

This brings up the next series of windows to let you set the various PIC and Pascal options; refer to **Figure 2**. We're in the **Project** tab now. Choose the particular PIC you've decided on from the pulldown menu, as well as the rate of the system clock. If you look carefully at the figure, you'll see I've selected the PIC16F88 running at 8 MHz.

Just leave everything else at default values for the time being. You can learn about this stuff later on, if and when you need to. Move on to the next tab (**Compiler**) as shown in **Figure 3**. The main item of interest here is the **Output Path** entry. You can have the resulting assembler code sent to any folder you want. (Just adjust the path by means of

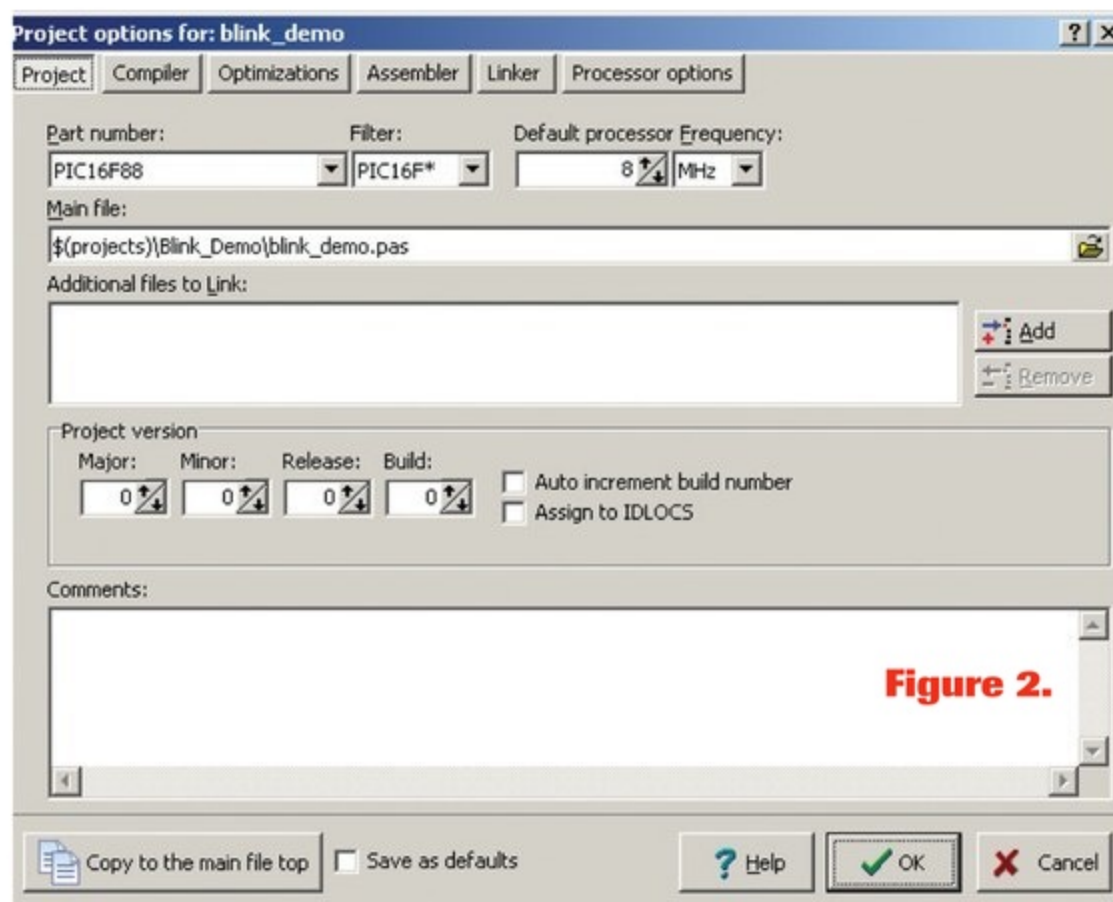


Figure 2.

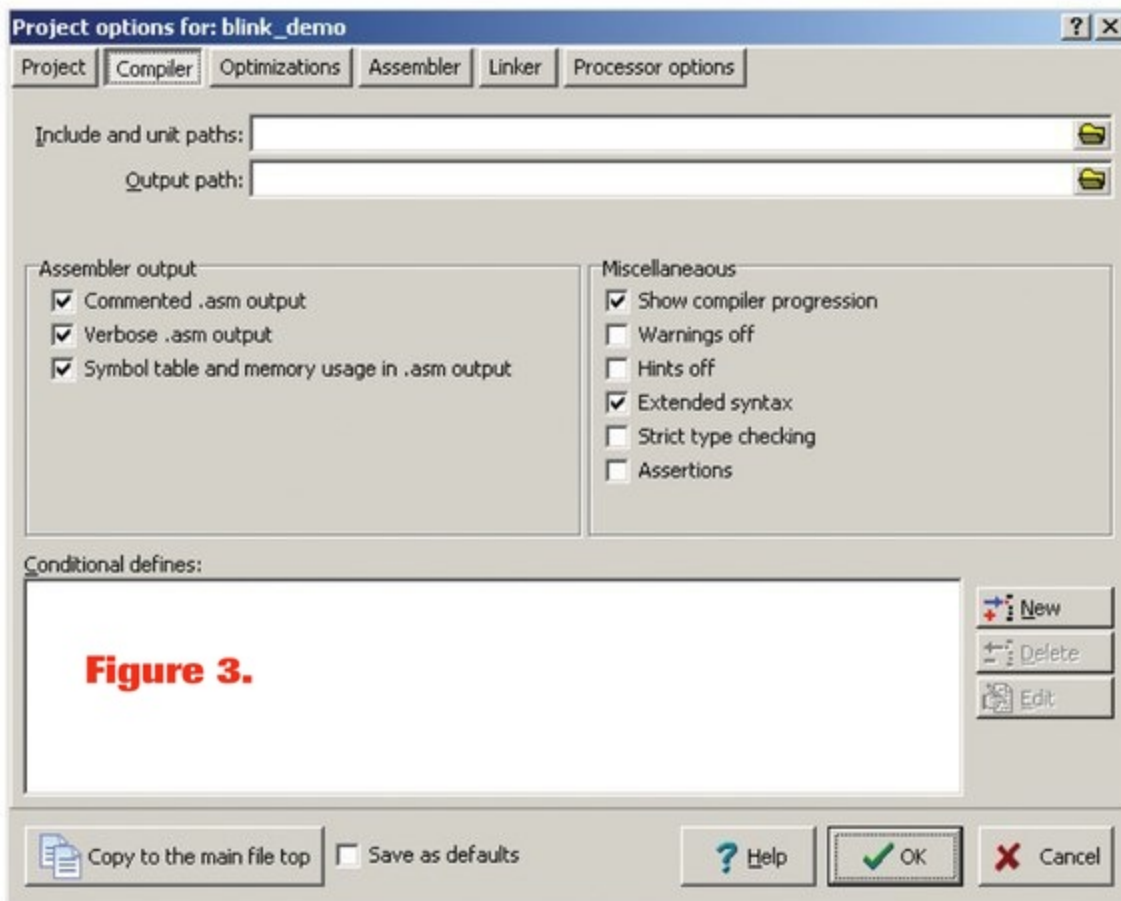
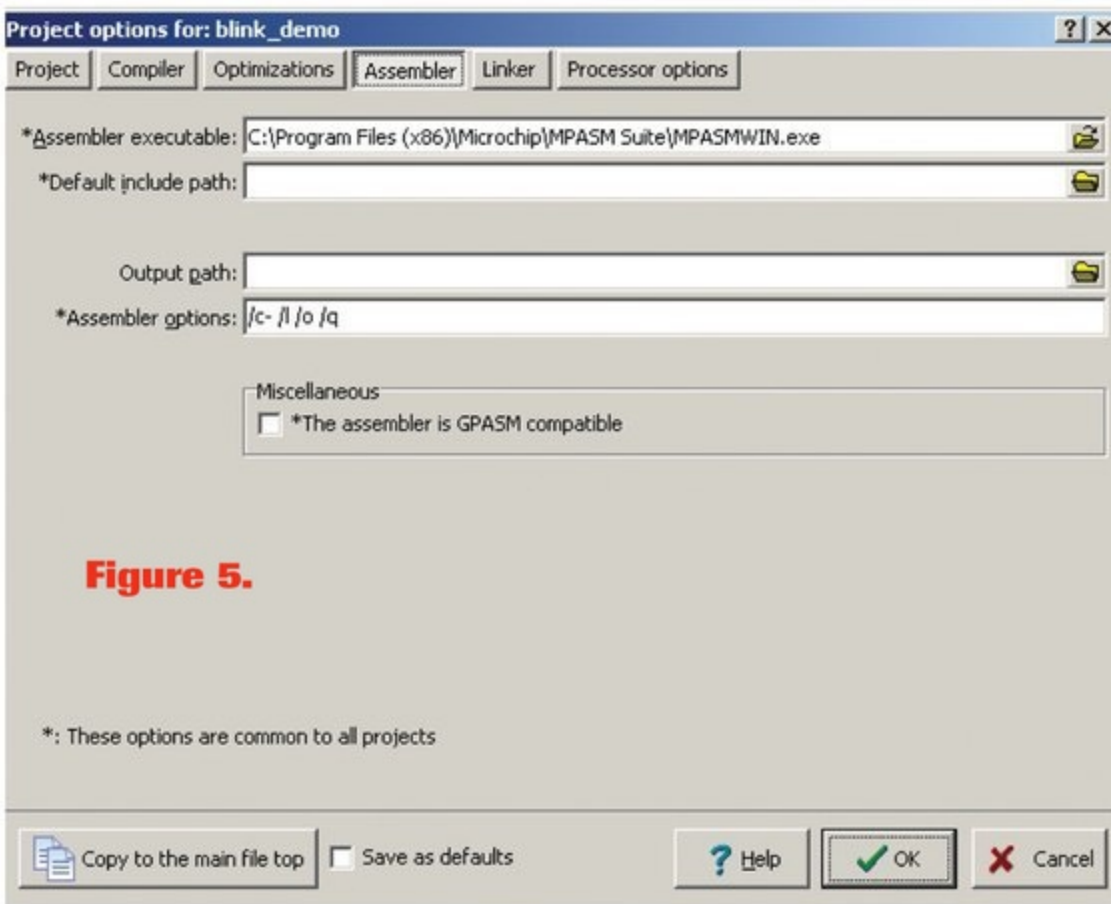
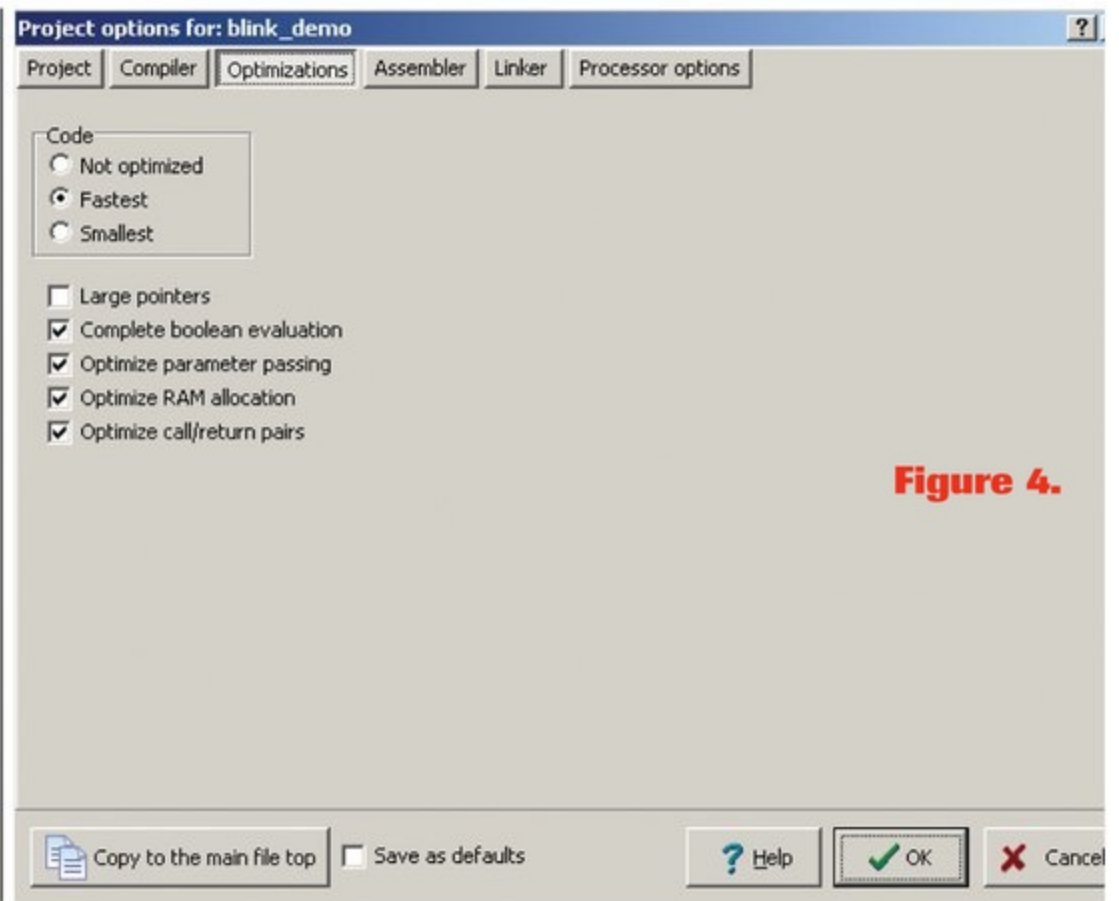


Figure 4.



the folder icon on the right). Generally, however, you'll simply want everything to end up in the project folder. Something that is rather confusing at first is that when the **Output Path** box is empty, the default project folder is used. Again, accept the defaults for any other options.

Next up is the **Optimizations** tab. I promised you I'd keep everything as simple as possible, so just go with the defaults there for now. Do come back to this at some later date because PMP has all sorts of powerful choices to optimize your code for speed, size, variable usage, and more. **Figure 4** shows what I saw when creating this sample program.

Move on to the **Assembler** tab next as shown in **Figure 5**. Here, you get a choice of which assembler to use. Since we began by installing the Microchip one, you ought to see MPASMWIN listed in the top box and its required command line parameters a little further down. As before, make sure the **Output Path** is blank, indicating we're

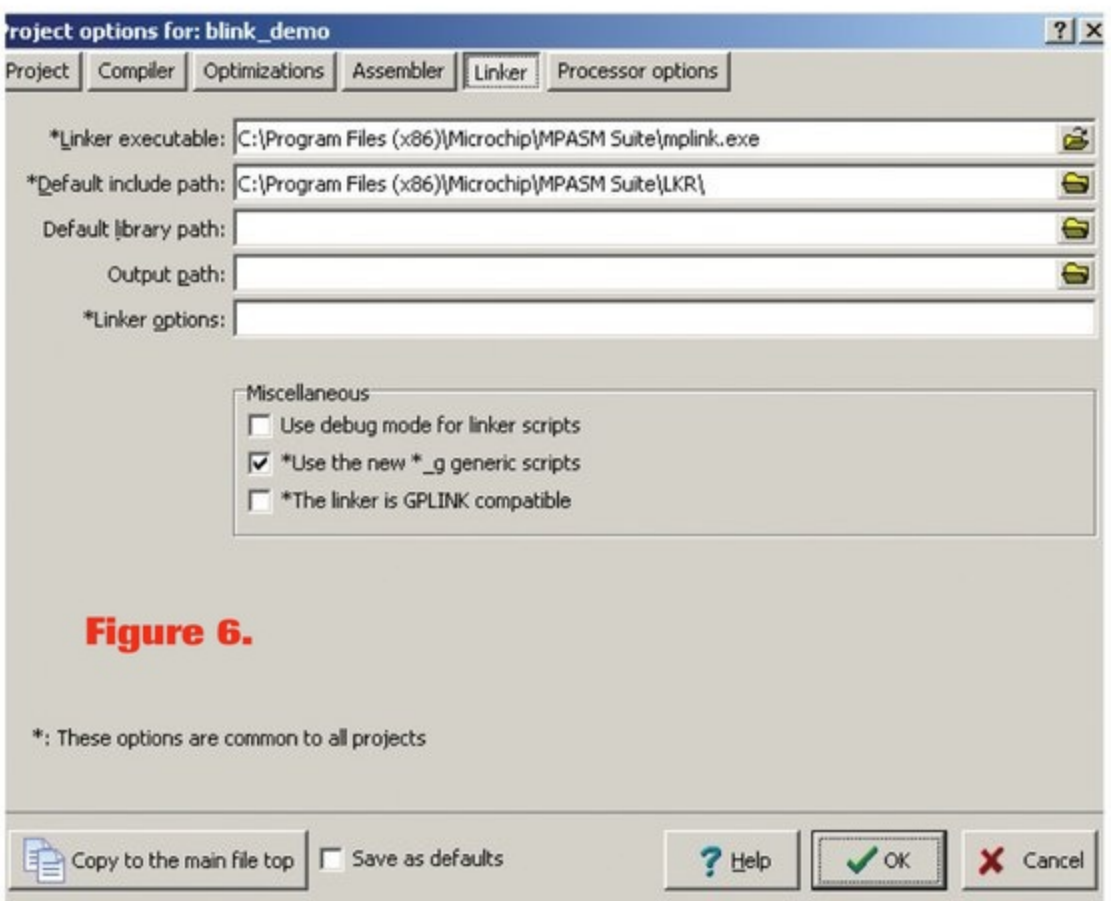
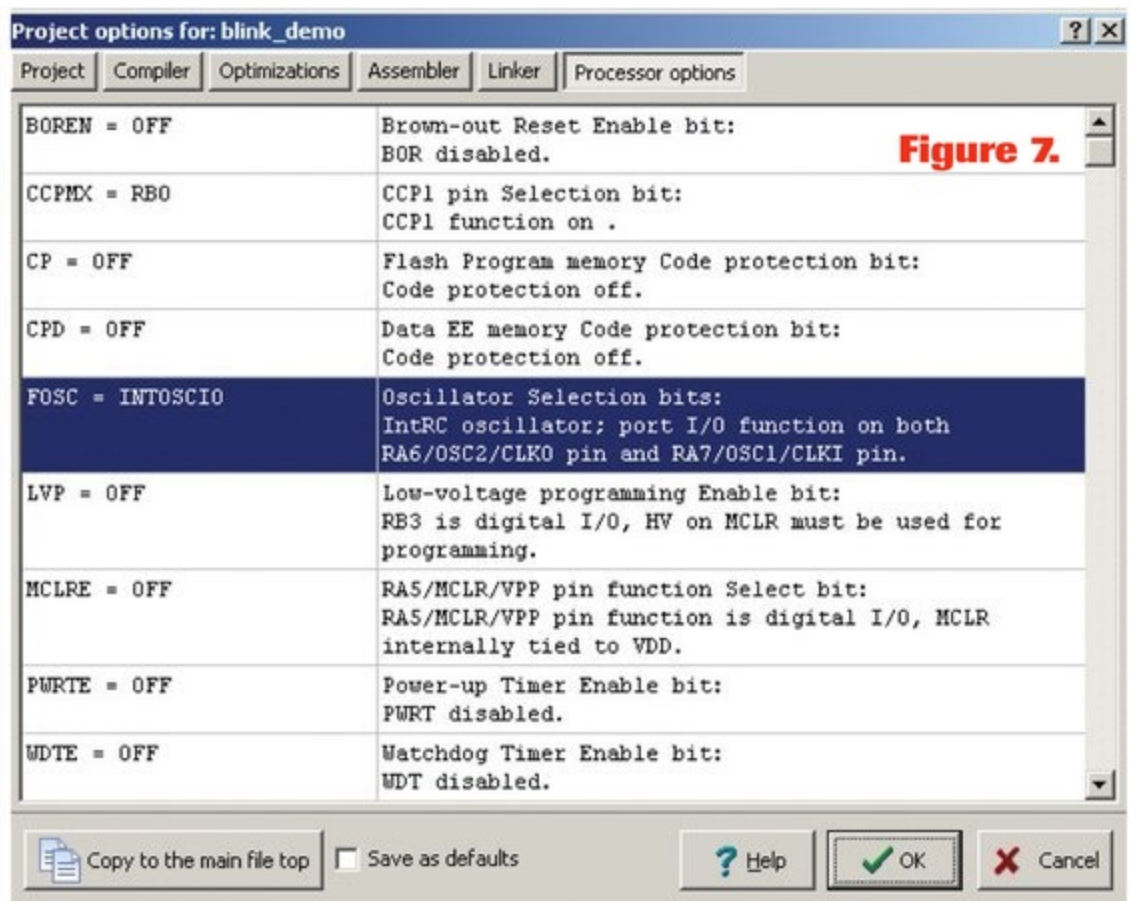


Figure 7.



happy to use the default path.

Next up is the **Linker** tab illustrated in **Figure 6**. You should be getting quite good at what's expected now, so make your choices if needed and move on to the last tab, **Processor Options** (refer to **Figure 7**). This is really cool!

Here, you'll find a menu of choices with explanations to set the configuration bits for the PIC you're using. PMP is smart enough to know what makes sense for your chip. To see the available options, simply right-click on each entry and then choose what you'd like. Among the more important options, I decided to go with the internal clock for the demo.

Then, the Main Program File

Now comes a bit of magic. In the lower left-hand corner of the **Processor Options** window, click the button entitled "Copy to the main file top." Bingo! A skeleton program is created, but more importantly all of those options you just selected are pasted into place automatically. Does that register? There's absolutely no need to untangle a weird alphabet soup when setting the configuration bits; you simply make your choices in the menus just described and PMP generates the required code.

Since we're creating our own project, delete all the sample program stuff between the BEGIN..END pair, and throw in a few carriage returns to make what remains neater. (Remember, in Pascal there's no penalty for extra white space like blanks, tabs, or carriage returns.)

Figure 8 shows what I wound up with after cleaning everything up. This might be a good time for you to save your progress so far.

Specify the Clock

Here's something that might trip you up without a helping hand; it sure had me scratching my head the first time around. Recall that we specified the processor speed to be 8 MHz earlier. Well, actually, that's just to let PMP know how to compute delay times internally; it doesn't actually set the clock speed within the PIC. To do that, we must store a value within the OSCCON register (at least for the PIC16F88 I'm using). I can almost hear you grumbling, "Great ... now I have to dig out the datasheet and look up the required bits."

Not to worry! A most helpful feature built into the PMP IDE (integrated development environment) lets you rapidly

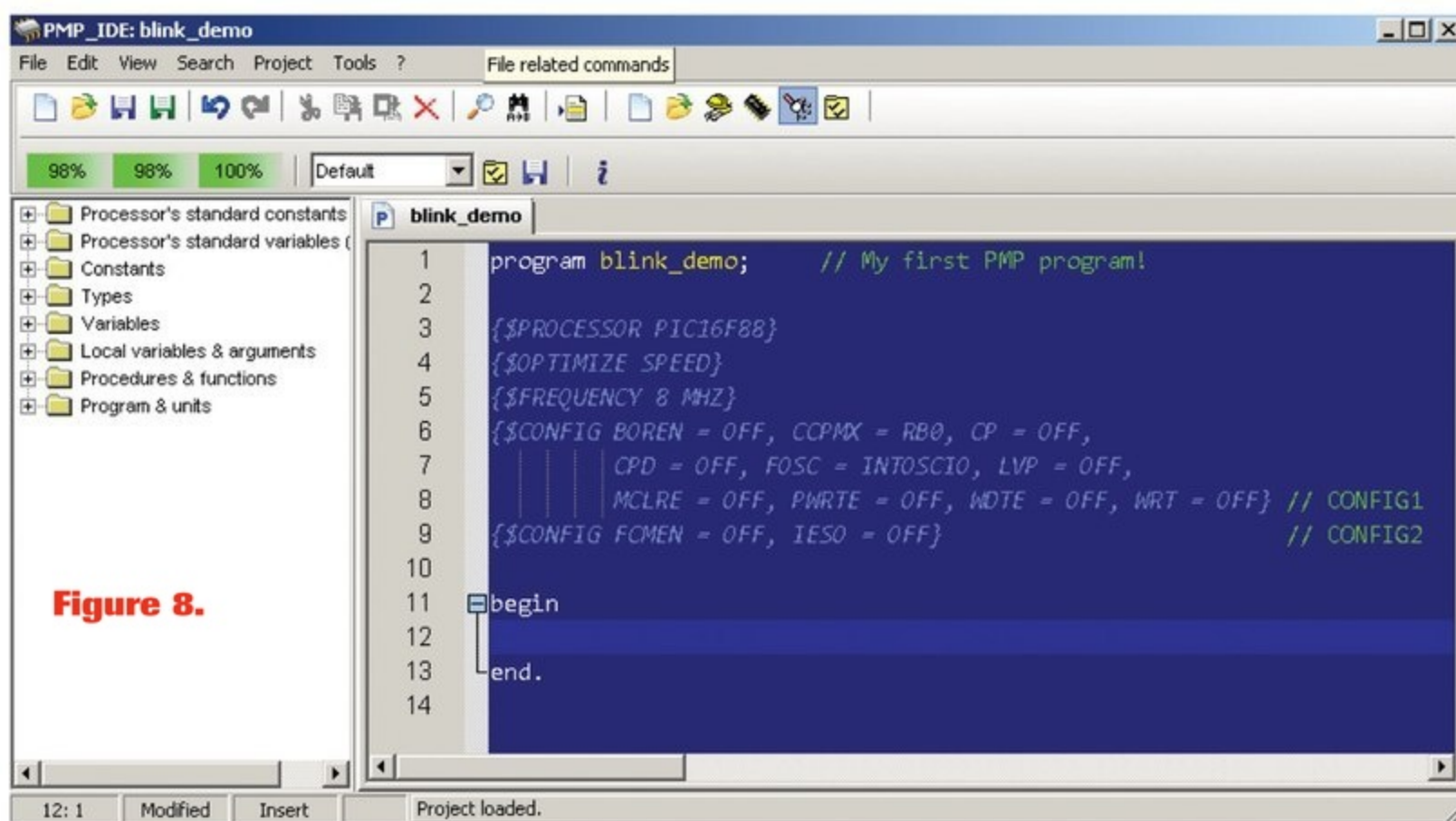


Figure 8.

look up details when you need them. Here's how.

From the menu bar near the top of the screen, choose **Project...Show Device Characteristics**. Bam! There's your pinout with descriptions, in case you've forgotten them (see **Figure 9**). Notice the hyperlink for the datasheet. Supposing this is your first time and that you're connected to the Internet, clicking this will load in the complete datasheet directly from the manufacturer and store it locally on your machine for future reference. It's all automatic.

The datasheet appears in a .pdf viewer for your inspection and you'll never have to load it again; it's become part of the IDE. I just looked it up myself, and could see that OSCCON needs to be stuffed with 0b01110000 to yield an 8 MHz clock. If all you need is a quick memory jog, then here's another great feature. From the menu bar, choose **Project...Show SFR Information** to see a brief summary of any register in question. **Figure 10** shows what popped up for me.

Let the Coding Commence

With that, we can start programming! (I eventually created a template for myself based on the various choices

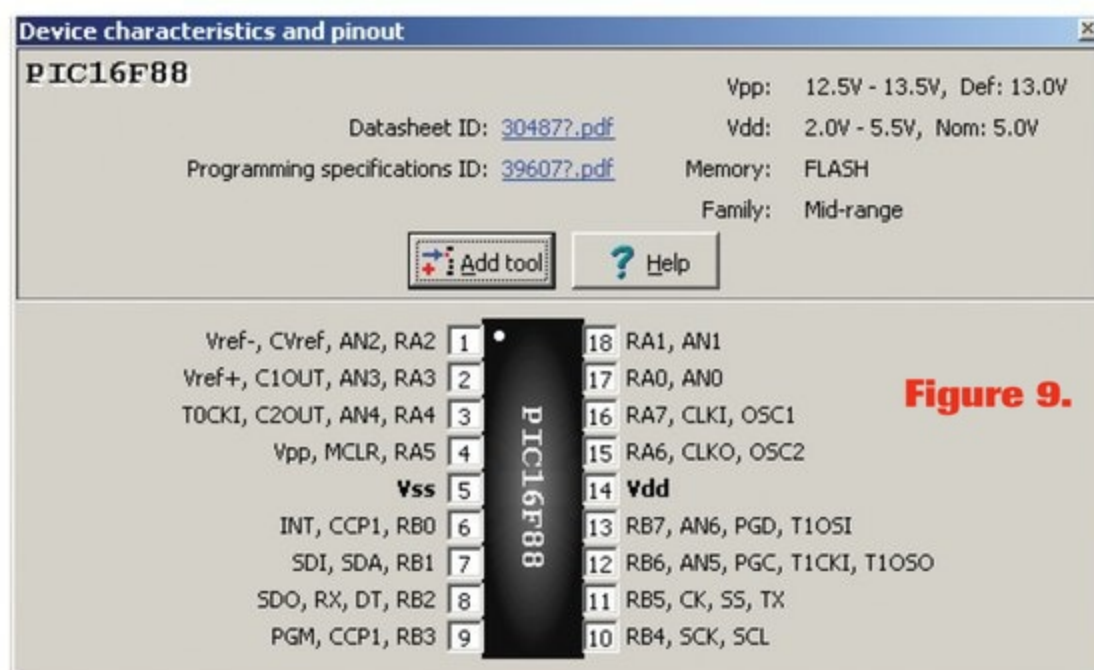


Figure 9.

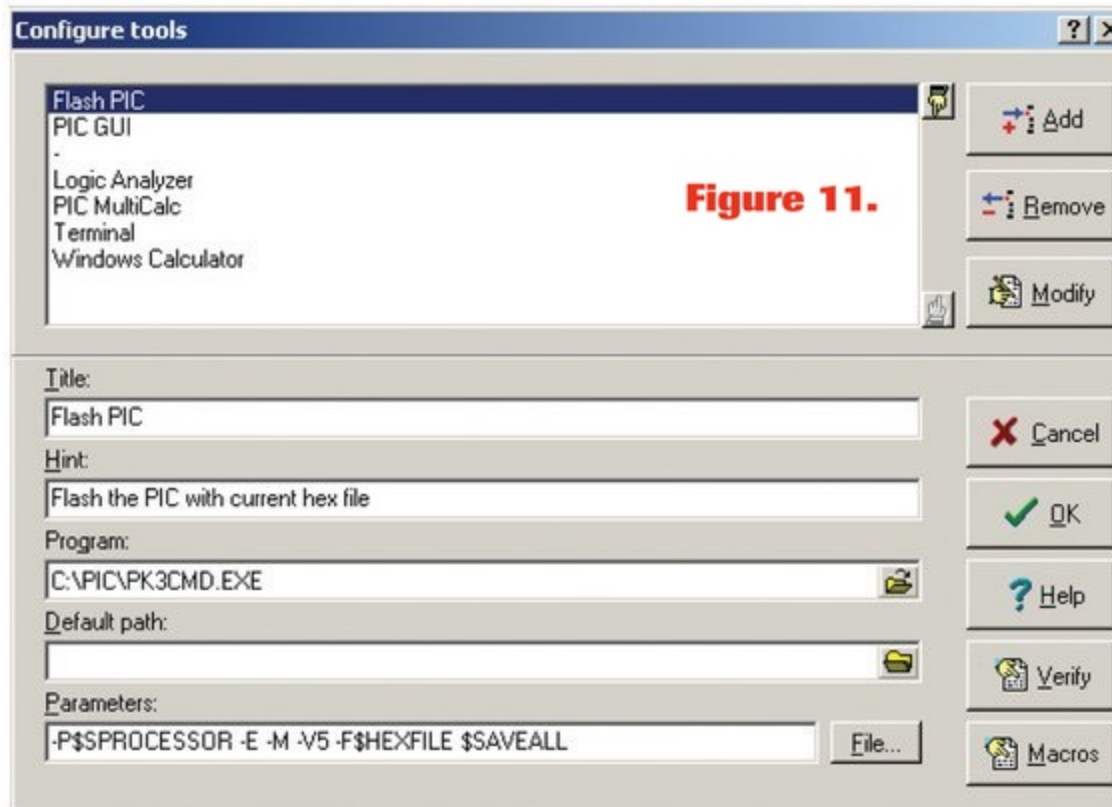
SFR information for PIC16F88

SFR: **OSCCON** Address: \$8F, Oscillator Control register

Figure 10.

	IRCF			OSTS	IOFS	SCS	
	IRCF2	IRCF1	IRCF0			SCS1	SCS0
Access	R/W	R/W	R/W	R	R/W	R/W	R/W
POR state	0	0	0	0	0	0	0
MCLR state	X	X	X	X	X	X	X
	B7	B6	B5	B4	B3	B2	B1
							B0

bother commenting on the actual code, other than to remind you that statements are separated by semicolons; compound statements are created with a `BEGIN...END` pair; comments are indicated by the double-slash; and that a period concludes the entire program. Don't forget that assignment to a variable is denoted by the colon-equal combination.



Time to Compile and Flash

We're now all set to build the code completely. Go to the menu bar once more and choose **Project...Build**. Without you lifting a finger, PMP compiles the code, assembles it, and then links it to create a final hex file ready to burn into the PIC. Assuming you have used the defaults throughout, you'll find the hex code in the project folder.

Before you can Flash (program) the PIC, you'll need to tell PMP what burner you're using and where its command files live. To do this, from the menu bar at the top of the editor, select **Tools...Configure Tools**, then choose **Add**. When I installed PICkit 3 on my computer, the configuration looked like that shown in **Figure 11**. Obviously, you'll fill in the boxes with whatever applies to your programmer. You'll notice that I've added in some other freeware tools which I've found useful when programming. Hence, they become part of my personalized IDE.

You're ready to Flash the program. Simply go to the menu bar again and choose **Tools...Flash PIC** (or whatever you labeled your customization as), and away you go! A few seconds later, you'll have a programmed chip all ready to start blinking that LED.

What's Next?

You should now be over the hump and ready to explore PIC Micro Pascal on your own. The editor is packed to the brim with all sorts of powerful features. There isn't enough space to describe them all here, so make it a point to try each one. Likewise, dig into the language itself. I think you'll be mighty impressed at the rich collection of commands, data types, and data structures within PIC Micro Pascal. **SV**

just made; that way, I can jump into the coding with no preamble.) Remember, our goal is to blink an LED on and off. For the chip I'm using, I'll hang the LED off port line B.0 through a 330 ohm current-limiting resistor. As we just learned, it's easy to confirm the pinout of the PIC you're using. In this case, all we need are B.0 and the two power pins. The required source code for the program body is a snap now:

```
begin
  OSCCON := 0b01110000; // set system clock to 8 MHz
  TRISB := 0;           // set Port B for output

  loop
    PORTB.RB0 := 1;      // turn LED on
    delay_ms(500);       // delay half a second
    PORTB.RB0 := 0;      // then off again
    delay_ms(500);       // delay half a second
  end
end.                     // repeat in perpetuity
```

There are plenty of Pascal tutorials out there, so I won't

Resources

PIC Micro Pascal (PMP) depends upon the assembler within Microchip's MPLAB, which is a complete integrated development environment (IDE) for assembly language programming. (Don't worry; you don't have to know assembler to use PMP). Begin by downloading and installing MPLAB. It's free, of course.

You'll find it at www.microchip.com/pagehandler/en-us/devtools/mplab/home.html.

Next, download and install PMP — also free of charge — from www.pmpcomp.fr.

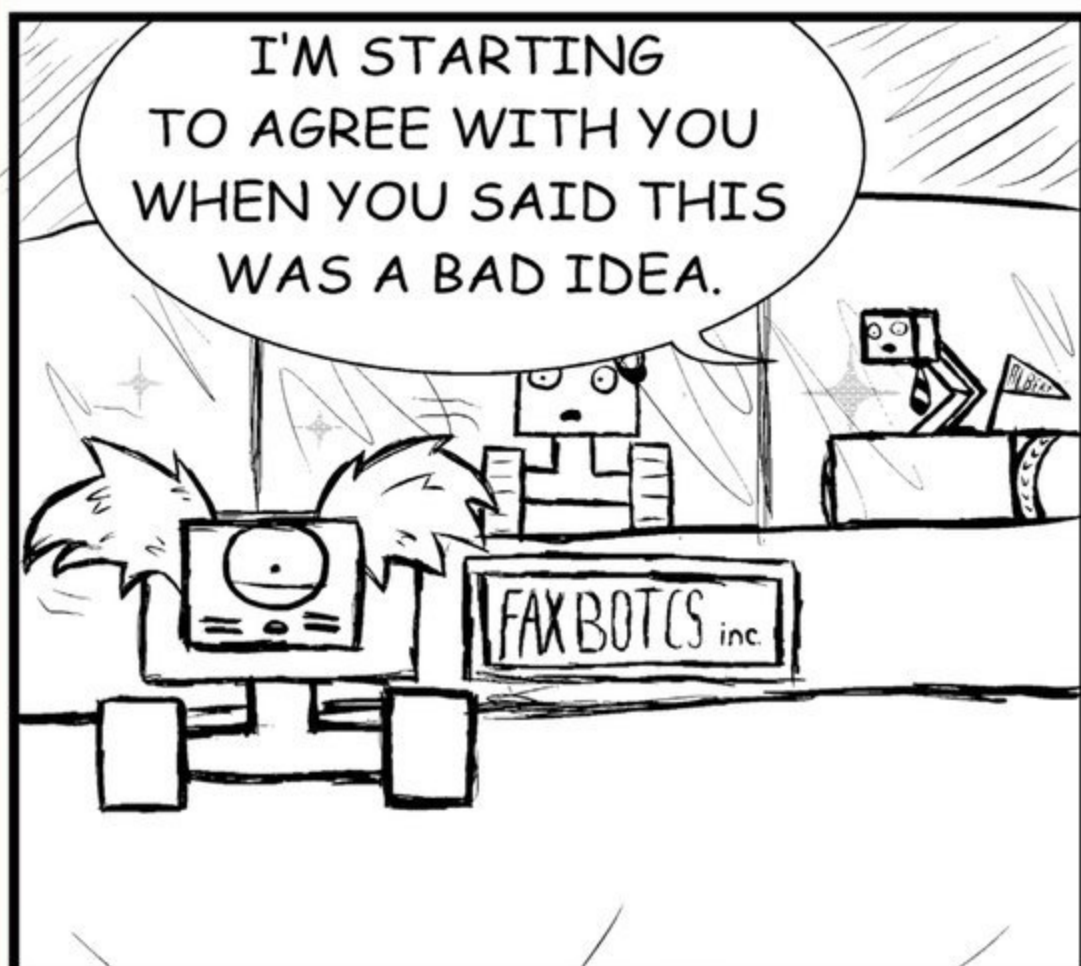
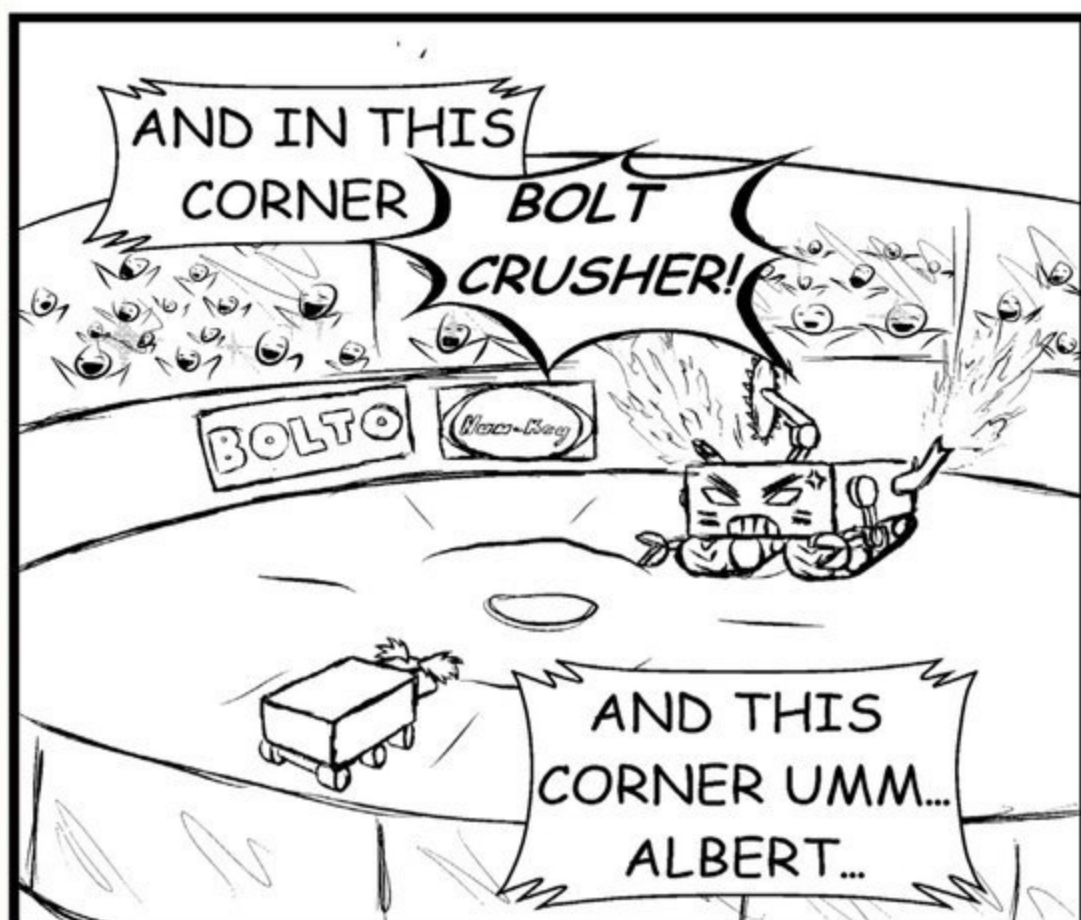
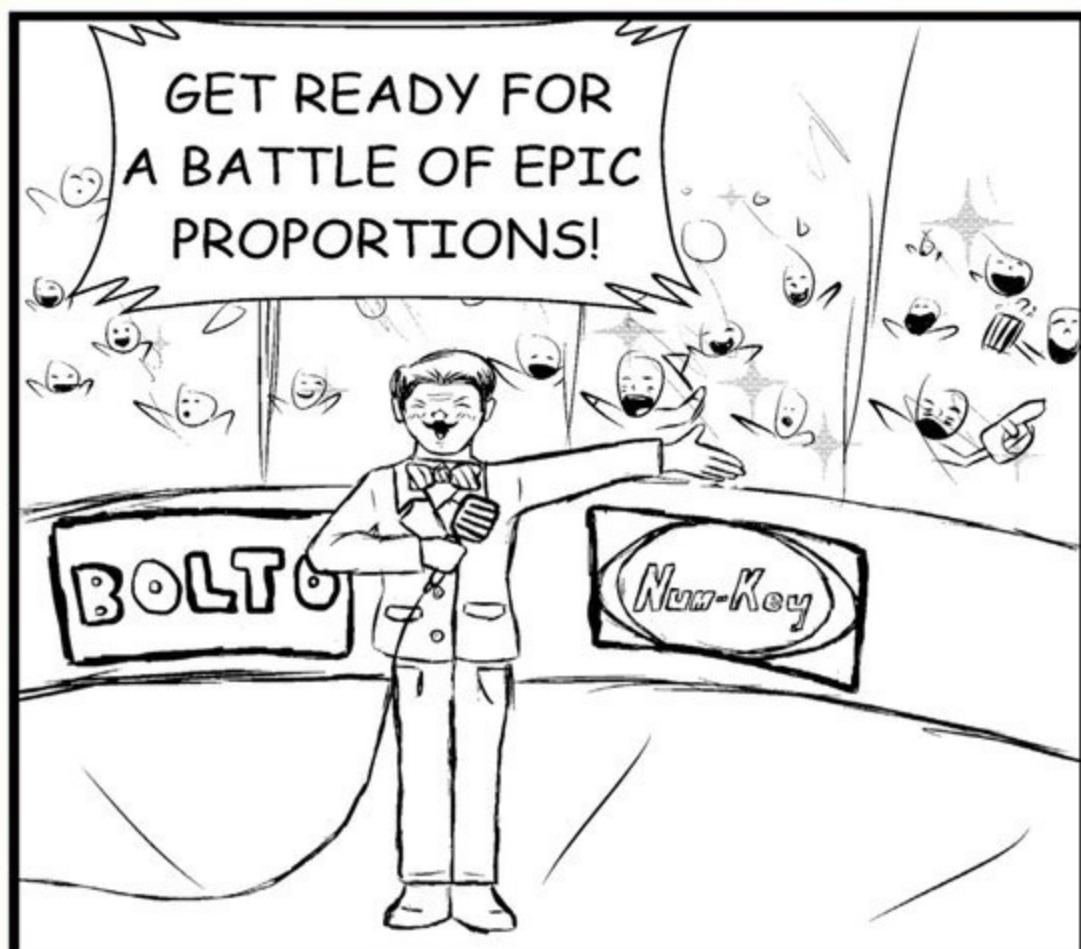
Installation is easy, and PMP takes care of searching out MPLAB on your computer and making the required connection.

You're all set to start programming! Remember, PMP comes with an excellent help manual which I recommend printing out right away. If you really do get stuck on something, consider joining the PIC Micro Pascal forum (yet again, free of charge) at www.pmpcomp.fr/smf.

I'm a frequent participant there myself and have contributed over 60 exercises to get you going with all sorts of applications: LCDs, LEDS, communications, music, sensors, motors, you name it.

MAXROBOTECH™

BY: AARON & ROBERT GROSS



EITHER WAY, YOU DO NOT LOSE!

Albert will lose this match. Yet he didn't fail. He will be rebuilt. Albert exits this contest with the joy of experience, learning, and of trying.

The experiences, the memories, the energy at the contest, and so many other things remain long after the contest ends. You, the robot builder will get more from entering the contest than you possibly know.

Robot contests changed my kids and my life. Nicole's robot lost a contest because it failed to detect the wall. This caused me to research sensors and later start a sensor company.

Later, Aaron's robot consistently won and Kristin's robot consistently lost. Kristin, frustrated, determined, and with my help added a micro-controller to her robot, changing it into a winner.

Loosing and winning robot contests gave us as a family the experience and fortitude in projects, and later the guts to move into business.

We enjoyed our local robot club contests, (even if it was a two hour drive). Two popular contests are robogames.net and robotbattles.com

© 2014

www.maxrobotech.com

Robots and Artificial Intelligence

The other day I was watching my cat, Rocky eating out of his dish in the corner of our kitchen which has a hardwood floor. After eating his fill, he began to use his paw to scrape up invisible dirt and cover his remaining food — much like a cheetah on the Serengeti Plains of Africa, covering a partially eaten gazelle. One can never be too careful when there are scavenging hyenas nearby, after all. I called out to my cat, "Rocky, you dumb cat! What do you think you're doing?"

Of course, he didn't know what he was doing and had no clue what I was saying. He was following his DNA that one of his ancient relatives learned to do to protect their food.

We might laugh at animal's antics, but psychologists certainly know of many crazy things that we humans do for reasons unknown.

Rocky's very small one ounce brain can only contain so much 'computational power' — much the same as a microcontroller in one of our robots. Yet, that small brain allows Rocky to do many amazing things.

Is intelligence overrated? Is artificial intelligence (AI) overrated? Can we ever hope to have sufficient

intelligence built into our robot creations in our lifetimes? Many universities, companies, plus robot and computer experimenters are trying very hard to make AI a reality for robots. However, many experts have a great fear of advances in artificial intelligence.

Artificial Intelligence as Viewed by the Public

It always seems that when there's something few of us really know much about is presented on TV or in movies, it becomes evil or malevolent — especially if it is of a technical nature. Many sci-fi movies seem to throw a twist into plots and distort science in a negative way. Robots and computers are good examples — especially when they are presented together.

The HAL9000 computer from *2001: A Space Odyssey* took over the spacecraft when it killed off three crewmembers because 'HAL' felt that the spacecraft's mission was more important than the crew's lives. Adding the glowing red eye as seen in **Figure 1** gave the computer an evil presence, as was done to the Terminator robot in **Figure 2**.

In the first *Terminator* movie, the main robot had a single purpose programmed into its 'brain' when it went back in time: kill the soon-to-be mother of the human resistance

leader of the future.

The same situation was evident in the Will Smith movie, *iRobot* when V.I.K.I. — the Virtual Interactive Kinetic Intelligence AI mainframe computer shown in **Figure 3** — directed all NS-5 consumer robots to turn against their human masters. Seeing how humans appeared to be in a self-destructing mode, V.I.K.I. felt that she had to protect humanity from being harmed by going against Asimov's Three Laws of Robotics.



Figure 1. The HAL 9000 AI supercomputer from the film, 2001: A Space Odyssey.

A History of AI

Thousands of books and articles cover AI in so many ways. Some go deep into coding or describe the types of computers used. There appears to be many definitions of artificial intelligence — much the same as it is



Figure 2. Terminator robot from the film, Terminator Salvation.

Post comments on this article at www.servomagazine.com/index.php/magazine/article/march2015_ThenNow.

for robots. I'm sure most of the readers of this magazine have a good grasp of the subject, so I won't go into any lengthy discussion of AI, but rather mention some key points of its history and a few issues that have caused a stir within the media.

One very good book by Pamela McCorduck entitled, *Machines Who Think* shown in **Figure 4** (written back in 1979), is one of the best that I have ever read about the history of artificial intelligence. Her use of the word 'who' in the title really made an impression on me.

I think of the movie, *Short Circuit* and the robot 'star' of the movie shown in **Figure 5** stating, "I am not just a machine; Johnny 5 is alive!" It is not that the robot became intelligent when struck by lightning as some movie descriptions state, it actually became *self-aware*. Adding a cute voice to the robot certainly made it more personable. This is a key point that many AI professionals feel is important in machine intelligence.

Let's discuss a few widely varied examples of AI as it is applied in robots and autonomous vehicles.

Robots with AI can Learn from Past Mistakes

Back in 2004, DARPA — the government's Defense Advanced Research Projects Agency — hosted the first of several autonomous vehicle



Figure 4. Pamela McCorduck's great book on Artificial Intelligence.

competitions that were held in real life outdoor environments. It was a well-planned approach for the military to define a set of goals to solve and allow competitors to compete with each other for prizes. This proved to be better than paying many millions out to several major companies ahead of time in hopes that they *might* arrive at a workable solution.

Their first DARPA Grand Challenge was an off-road 150 mile course in the desert between California and Nevada in March 2004. The goal was for the military to ultimately have a third of their ground forces be autonomous by 2015.

With this in mind, Congress authorized DARPA to offer a \$1 million prize for the winner. One hundred teams entered, and the closest that anyone got to completing the course was just 7.32 miles. That was Carnegie Mellon University's modified Hummer (shown in **Figure 6**); it got hung up on a rock while

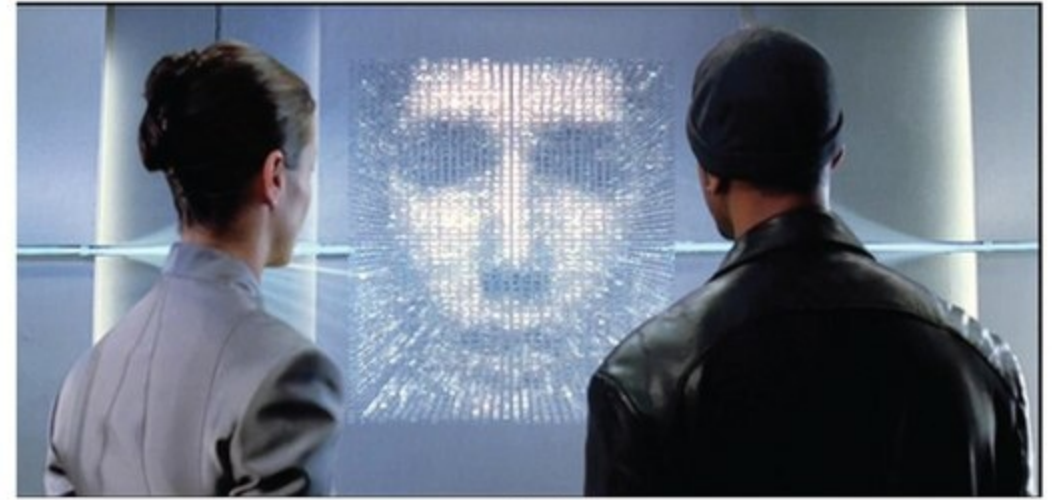


Figure 3. Image of the AI supercomputer V.I.K.I. from the film, *iRobot*.

making a sharp turn. Others ran off the course, got hung up on a fence, or just died. There was no winner.

Was this an embarrassing mistake for DARPA? Absolutely not! More is learned from real life mistakes than ever from computer simulations.

A year and a half later, the second Grand Challenge was run and of the 23 starters, all but one made it past the original 7.32 mile mark of the last race. The results were a tremendous improvement over the previous contest.

Five vehicles finished the race, with 'Stanley' of the Stanford University Racing Team winning in six hours and 54 minutes. Carnegie Mellon University's two entrants placed second and third at 7:05 and 7:14.

The competitors had to pass through three narrow tunnels, a hundred tight turns, and a winding mountain pass with a sheer drop-off on one side. Not bad for off-the-floor vehicles (modified, of course) with no



Figure 5. Robot Number 5 from the film, *Short Circuit*.

Figure 6. CMU's Modified Hummer, Sandstorm in the 2004 DARPA Grand Challenge.

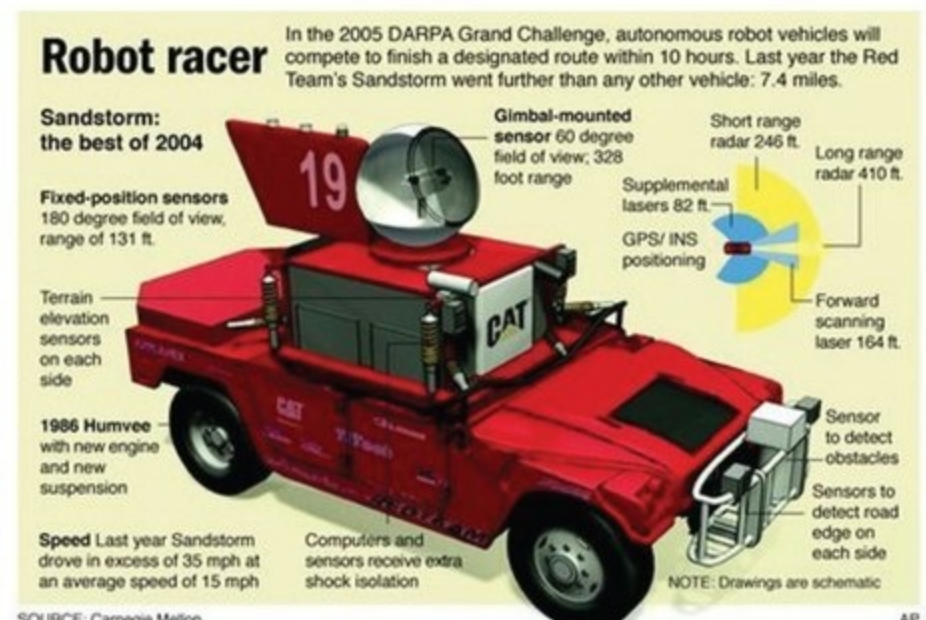


Figure 7. Sensor close-up from the DARPA Grand Challenge.



human drivers. The race included a huge 30,000 pound Teramax vehicle from Oshkosh Truck that finished in fifth place.

The “robots” themselves did not learn from the previous contestant’s mistakes; the coders used the mistakes to write better code.

What is Artificial Intelligence?

I have used that phrase here several times, but what exactly is artificial intelligence? I have browsed many sites and few really begin with a definition of the science. Most seem

to assume that a reader already has their own idea of what AI is, what it consists of, and how it is characterized. Wikipedia defines it this way:

Artificial intelligence is the intelligence exhibited by machines or software. It is an academic field of

study that studies the goal of creating intelligence. Major AI researchers and textbooks define this field as “the study and design of intelligent agents,” where an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success. John McCarthy, who coined the term in 1955, defines it as “the science and engineering of making intelligent machines.”

DARPA Robots

Considering that description of AI, would we call the entrants in the DARPA Grand Challenges truly intelligent? These vehicles had multiple microprocessor-based computers and were covered with multiple sensors as shown in **Figure 7** ... but does that qualify them as intelligent?

How about the robots that were developed for the DARPA Robotics Challenge that were intended to be disaster relief robots, like the MIT developed Atlas shown in **Figure 8**? This competition is the most unique and costly of the DARPA challenges and will be among the first contests to exhibit robotic artificial intelligence.

“The DRC is a competition of robot systems and software teams vying to develop robots capable of assisting humans in responding to natural and man-made disasters. It was designed to be extremely difficult. Participating teams — representing some of the most advanced robotics

research and development organizations in the world — are collaborating and innovating on a very short timeline to develop the hardware, software, sensors, and human-machine control interfaces that will enable their robots to complete a series of challenge tasks selected by DARPA for their relevance to disaster response.”

Looking a bit deeper at these DRC robots, there were 19 teams that qualified through several trial sessions beginning back in 2012, and it will end this June with the winner taking home \$2 million. The NASA JPL-developed RoboSimian shown in **Figure 9** is just one unique entrant in the competition. Six of the participants were awarded Atlas robots built by Boston Dynamics for DARPA as development platforms at a cost of \$10.9 million. This robot is shown in three poses in **Figure 10**.

The finals of the competition series will take place June 5-6, 2015 in Pomona, CA. The robots will be required to perform a series of complex and physical tasks with minimal telecommunications from human operators.

Though changes are expected, the initial task requirements for robot entries are:

1. Get into and drive a utility vehicle at the site.
2. Walk over a series of obstacles.
3. Climb a 60 degree ship’s ladder.
4. Remove blocks of wood and pipes.
5. Walk through three different types of doors.
6. Turn on a drill and cut an opening in a wall.
7. Mate a hose to a spigot.
8. Turn three different industrial valves.

In addition to these, contestants will be required to solve another unknown task at the site.

The Atlas robots stand six feet tall, but weigh more like a Sumo wrestler at 330 pounds thanks to titanium and aircraft-grade aluminum

Figure 8. Atlas DRC robot developed by MIT and Boston Dynamics for DARPA.



and some heavy hydraulic systems. The robot is equipped with two vision systems, a set of stereo cameras, and a laser range finder. The legs and arms consist of 28 degrees of freedom with the hands capable of basic dexterity.

Originally powered by external electrical and hydraulic power through a tether, the tether will be removed by the final contest. Internal computers control all motions and read the many sensors on the robot's body. Atlas is also capable of withstanding being hit by projectiles and balancing on one leg. In 2013, DARPA program manager Gill Pratt compared the prototype version of Atlas to a small child, saying that "a one year old can barely walk; a one year old falls down a lot ... this is where we are right now."

A key description of the competition and the robot competitors is that they are human-supervised ground robots capable of executing complex tasks in dangerous, degraded, human-engineered environments. Looking at it in another way, think of the Fukushima Daiichi power plant in Japan destroyed by the tsunami in 2011. Radioactive materials heavily contaminated it and hydrogen explosions had damaged many parts of the buildings. The environment was constructed for humans with doors, ladders, and human-sized passageways.

From a robot's point of view, all of the controls, valves, and lighting were designed for humans using human tools. Much debris was spread about and made walking, traversing, and working difficult. Many different teleoperated robots are still being used in this very difficult cleanup task.

Now, design a human-sized robot with human-like appendages to work in that or similar environments with almost constant human supervision. Allow autonomous operation when outside human monitoring has ceased for unknown reasons. Looking at

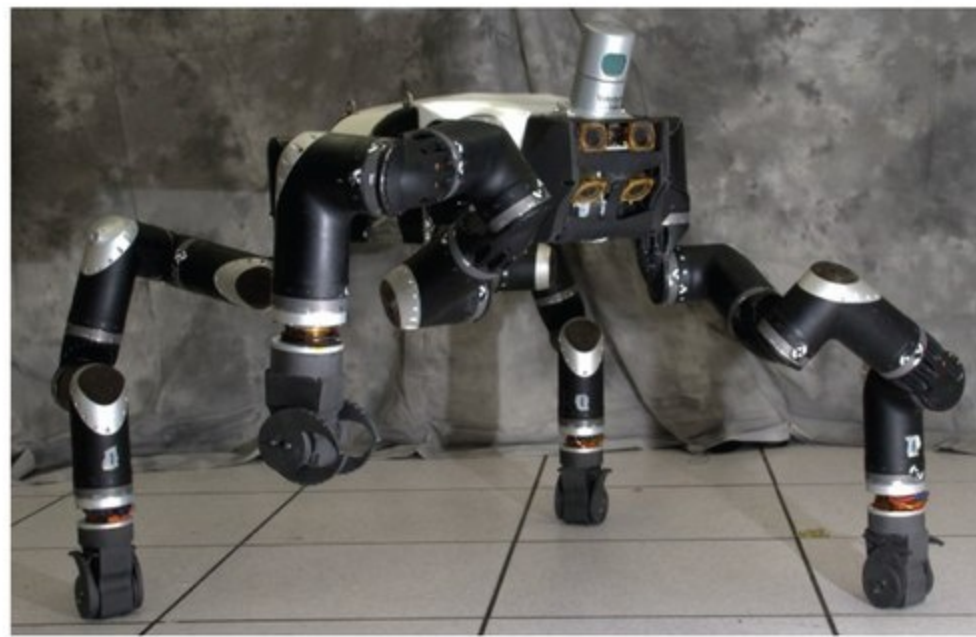


Figure 9. NASA JPL-developed RoboSimian 2.

these robots and what they are asked to perform, is this considered robot artificial intelligence? I think so.

The Evolution of RoboCup Robots

Stepping aside from the \$1.8 million DARPA Atlas robots, let's examine the more affordable RoboCup robots designed to play soccer. It all started back in 1997, and has progressed quite a bit in the past 17 years.

It was 1997 when the Deep Blue supercomputer built by IBM beat world champion chess player, Garry Kasparov in six chess matches.

Attendees at the International Joint Conference on Artificial Intelligence in Nagoya, Japan concluded that computers could also beat humans in *physical* competitions as intelligent robots. In fact, by 2050, robot experts at the annual world robotics championship hope to create a team of robots that can best the winning FIFA World Cup team in that year. (We are talking world class *human* soccer players.)

Robots in all sorts of applications had been discussed for many years. A professor from the University of British Columbia wrote a paper entitled, *On Seeing Robots*. Alan Mackworth felt that a successful robot soccer team could solve several nagging robotics problems such as cooperative team actions and the ability to follow a randomly moving ball. Mackworth and his team at UBC launched the Dynamo Project: the world's first attempt at an autonomous robotic soccer team.

The Dynamo Project carried out a series of successful experiments from 1992 through 1994, and what many people felt led to what is now the RoboCup competitions. The first event

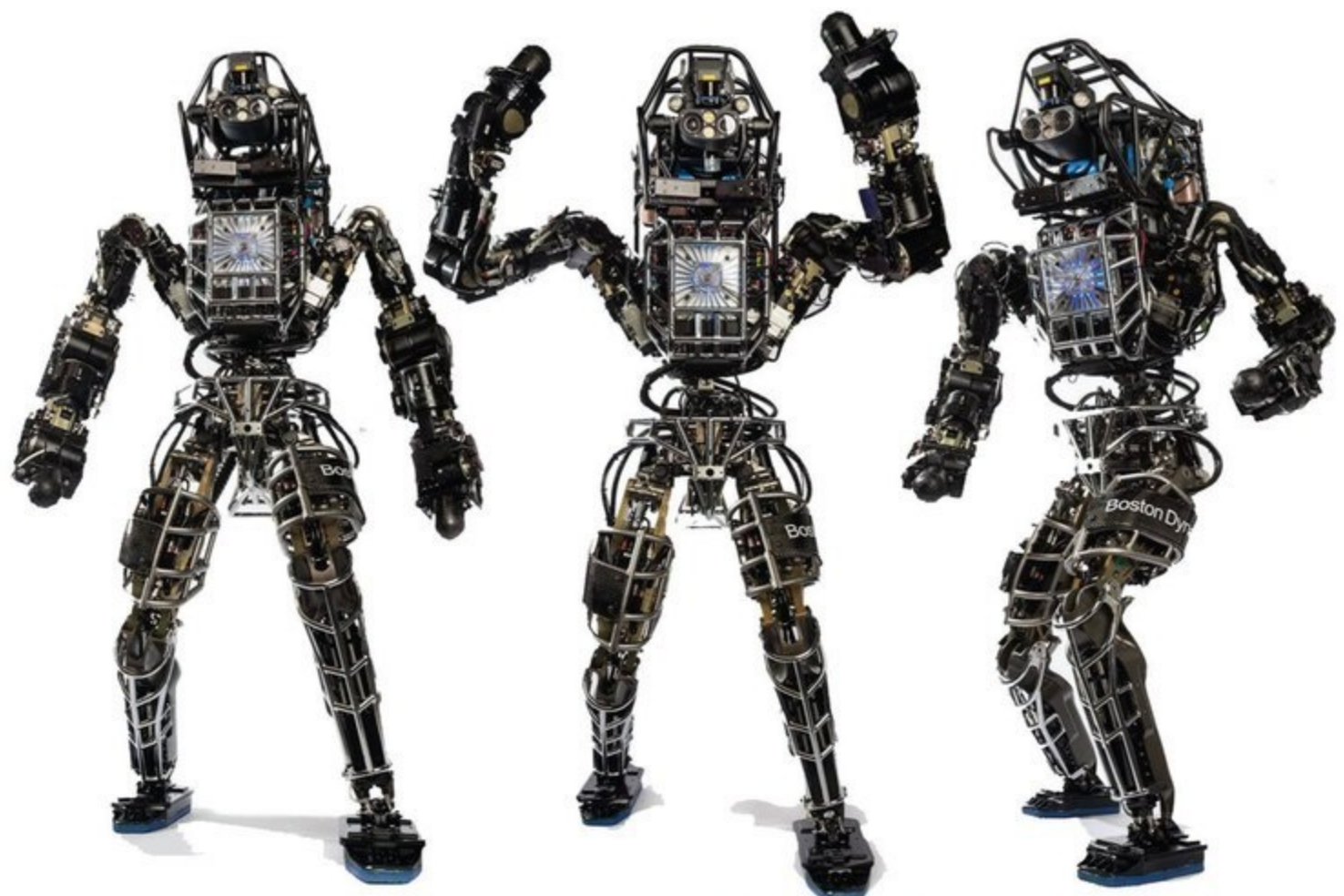


Figure 10. DARPA Atlas DRC robot in three poses.

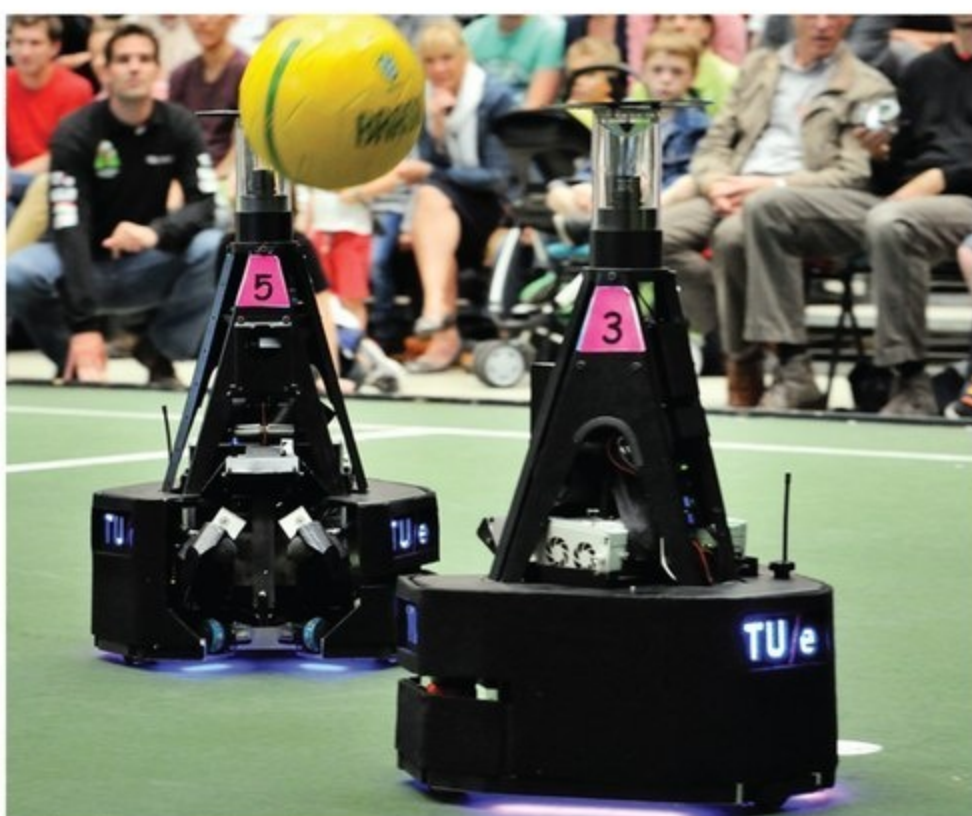


Figure 11. Middle Size League demo match in Eindhoven.



Figure 14. Google's Larry Page and Sergey Brin in Prius, with previous CEO, Eric Schmidt.



Figure 13. Aldebaran's NAO humanoid robots — a new RoboCup category.



Figure 12. Two humanoid RoboCup robots in competition.

— held in Japan — had 38 teams from 11 countries. The 2013 competition (held in Eindhoven, The Netherlands) had 2,500 participants from 40 countries, with 40,000 people watching the action. Not only do these competitions draw AI researchers and robot 'gearheads,' but even royalty, as Queen Máxima of the Netherlands was in attendance.

A scene from the 2013 competition is shown in **Figure 11**. Two 'Middle Size League' robots are handling a ball in a demonstration for the crowd. Not only do these robots possess advanced vision systems with visual motion processing capabilities, but you can also see the ball grasping and tossing mechanism on robot #5. The robots continually watch the moving ball, as well as their own constantly moving teammates and their opponents, and have to process all this information very quickly in order to play the match. This is an

excellent application of AI with robots. Two different humanoid-style robots are shown

in **Figure 12**. Aldebaran's NAO shown in **Figure 13** has replaced the no-longer manufactured Aibo for the RoboCup competitions. Costing about \$15,000 when originally offered (the new price is \$7,990), it is now in the cost category of the \$12,000 Robotis DARwin OP Deluxe (shown in **Figure 11** on its knees).

These are remarkable robots with sophisticated vision and facial recognition. Hopefully, by 2050 we will see robots with true AI capabilities that will challenge human soccer players in the World Cup ... or at least give them a run for their money.

Google's Self-Driving Cars

Most of us have probably seen the Google cars with the panoramic camera on the top for many years driving down streets and highways, documenting the terrain for Google

Earth. Google had been experimenting with self-driving cars for a while now — first with a Prius shown in **Figure 14** with co-founder and present CEO, Larry Page in the 'driver's seat,' co-founder Sergey Brin behind Page, and Eric Schmidt, CEO from 2001 to 2011 behind the car.

The new car prototype shown in **Figure 15** reminds me of the old BMW Isetta that I used to have years ago. The Google car program is lead by former Carnegie Mellon researcher and DARPA Grand and Urban Challenges team member, Chris Urmson. Urmson is first developing the car for Google employees, and then as a public service for the city of Mountain View, CA — home of Google.

The prototype operates without a steering wheel or pedals. Testing on public roads should begin this year, however, the cars will need to have a steering wheel and pedals for emergencies to satisfy California DMV regulations. Google supposedly has mapped every road in the city and driven all of them in autonomous



Figure 15.
Google's self-driving car prototype.

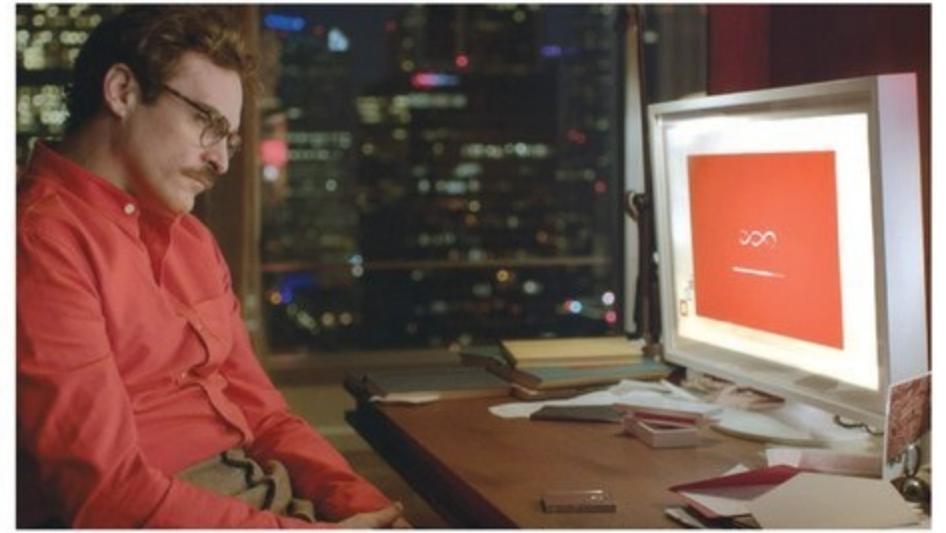


Figure 16.
Joaquin Phoenix in scene from the film, *Her*.

mode, with some external chaos thrown in for good measure.

Testers of Google's cute little self-driving cars can address a 'random mode' to go to a particular destination, Urmson said. "The car has encountered things it might not otherwise have seen, such as piled-up leaves, stray garbage cans, and, in one case, a woman in a wheelchair chasing a duck while wielding a broom. We would not have had a woman in a wheelchair chasing a duck with a broom as a test case," he said. "Fortunately, the vehicle did fine."

Wired Magazine has described the car as "an egg with the face of a koala." Trust in an autonomous car is still a big factor, and **insurance.com** has stated that only 25 percent of people surveyed would allow such a car to drive their children to school. Google is not the only company looking into autonomous cars as several other groups worldwide also have prototypes in the works.

Applying AI to Robots

Some of you might have seen the film, *Her*, starring Joaquin Phoenix (seen in **Figure 16**) as Theodore who falls in love with the voice of Samantha in the new 'OS1' software package he installs on his computer. Recently divorced, his friendship

progressed from 'just friends' to love with the powerful AI program's soothing voice that helps him in many ways in his life.

Now, add such an

intelligent and insightful AI program to a full-sized humanoid robot and you might have a character like Chappie, from Sony Pictures' film of the same name, opening in theaters March 6th. A scene with the robot is shown in **Figure 17**.

In the film, Chappie was 'kidnapped' and later adopted into a rather strange human family. The very intelligent robot had to grow up and learn his place in the world of humans. Chappie is: "The first robot with the ability to think and feel for himself. His life, his story, will change the way the world looks at robots and humans forever."

Final Thoughts

Researchers are just touching the surface of artificial intelligence and

have a long way to go. Sensors for robots are becoming less expensive and more capable, but are yet to truly approximate our human brains and amazing eyes that are far more than just image sensors.

We marvel at Siri for our iPhones, or Kinect for our Xboxes. Even IBM's Watson seemed amazing as it beat Jeopardy's two best players, but each of the reverse questions had to be previously typed into the computer and then entered into the computer at the same time as the verbal question was spoken to the two human contestants.

Yes, we are making great strides in AI and synthetic intelligence. However, there remains many more advances before we arrive at that point in AI where we are truly "summoning the demon," as Elon Musk is warning us.

Hugh Jackman's character, Vincent from the movie, *Chappie* may have hit the nail on the head when he states in the movie, "The problem with artificial intelligence is it's way too unpredictable." **SV**

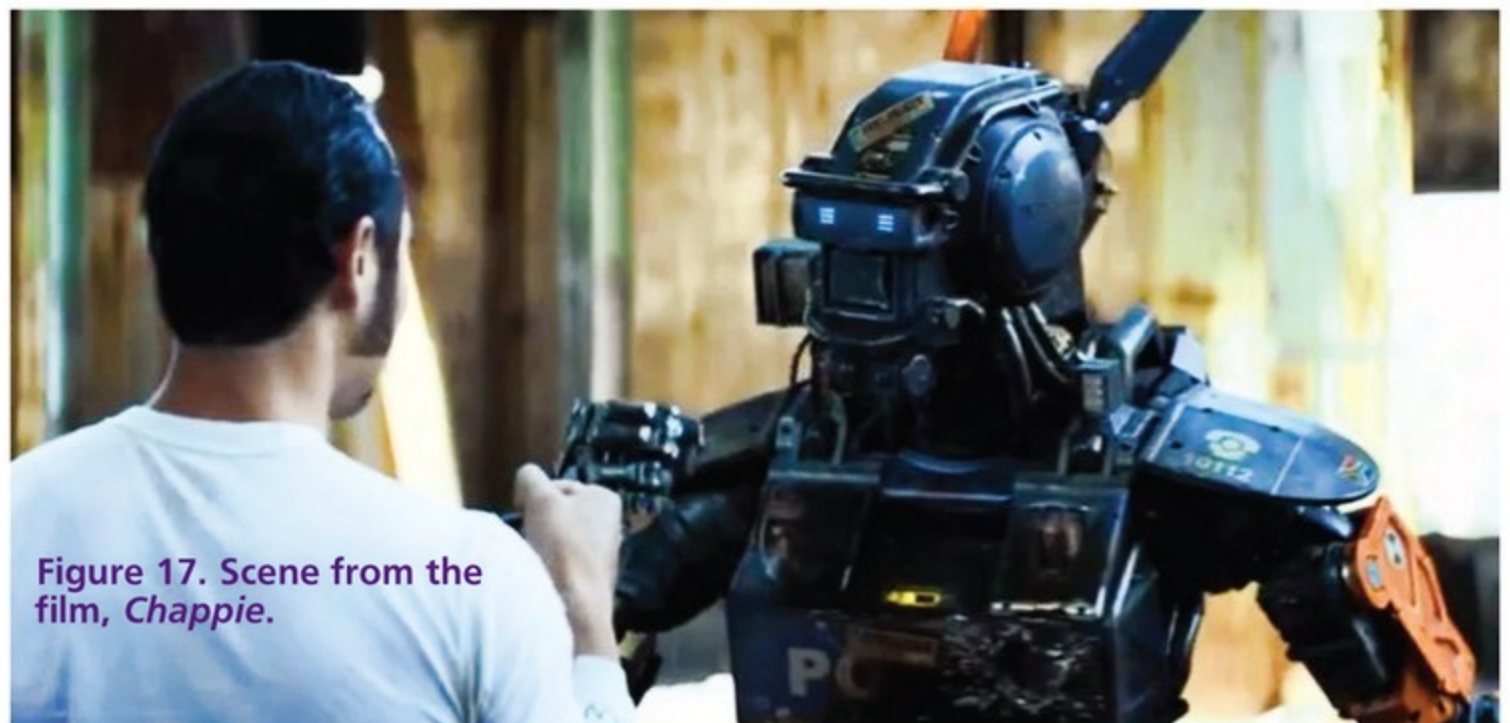
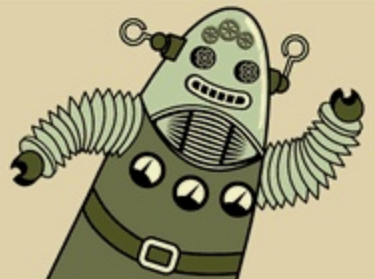


Figure 17. Scene from the film, *Chappie*.

ROBO-LINKS



MaxBotix
High Performance Ultrasonic Rangefinders

www.maxbotix.com
info@maxbotix.com Phone: 218-454-0766

RobotShop
.com



SERVOCITY TM \$299.99

6-14V HEAVY DUTY

Linear Servos

Three speeds to choose from!



to controller
to battery

Plug & Play

Need electronics?
www.nutsvolts.com

superbrightleds.com

COMPONENT LEDs • LED BULBS • LED ACCENT LIGHTS



AndyMark
Inspiring Mobility



www.andymark.com

For the finest in robots,
parts, and services, go to
www.servomagazine.com
and click on **Robo-Links**.

INVEST in your BOT!



HiTEC

12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecusa.com

10 CD-ROMs & Hat Special



Free Shipping!

Only \$ 208.95
or \$24.95 each.

ADVERTISER INDEX

All Electronics Corp.	21	Parallax	35
AndyMark	17, 82	RobotShop, Inc	61, 82
AUVSI	16	RoboGames	52
CH Robotics	9	SDP/SI	21
Firgelli	13	Servo City/Robot Zone	82, 83
HiTec	2, 82	Sherline Products	3
LF Components	69	SuperBrightLEDs	82
littleBits Electronics	Back Cover	Team Synergy Moon	48
Maxbotix	82	The Robot Marketplace	28
Maker Faire	49	Tormach	15
PanaVise	28	Weirdstuff Warehouse	21

GEARMOTORS

Put your project in motion!

\$24.99

3-12V Standard

\$34.99

6-12V Precision

\$27.99

3-12V Precision Planetary

\$39.99

6-12V Precision Planetary

\$9.99

6-12V Micro

NEW

3-12V Planetary Gearmotor with Encoder! \$49.99

4mm D-shaft

- Dual Ball Bearings
- Full Metal Gears
- 3PPR Magnetic Encoder
- 11 different speeds

6-pin JST connector

Tons of compatible accessories

12V LINEAR ACTUATORS

\$129.99



12V Heavy Duty Linear Actuator
- offered in 6 stroke lengths: 2"-12"
- aluminum stroke rod & outer tube
- metal drive gears
- choose from 25, 115 or 180 lb thrust!

\$399.99



12V Super Duty Linear Actuator
- offered in 6 stroke lengths: 4"-24"
- stainless steel rod & outer tube
- precision ball screw drive

NEW



NOMAD OFF-ROAD CHASSIS

\$279.99

MOTOR MOUNTS

Our growing selection of aluminum motor mounts make it easy to find the perfect mount for your application!



Digital Manual Speed Controller

\$79.99

DMSC6-16-10

+ MOTOR CONTROLLERS -

\$69.99

2x 5A

\$89.99

2x 15A

\$124.95

2x 30A

RoboClaw Motor Controllers

\$199.99

2x 60A

Multi mode interface can be controlled from standard RC radio, serial devices, analog and or microcontrollers such as an Arduino or Raspberry Pi.

www.SERVOCITY.com

Copyright 1999 - 2015 Robotzone, LLC - All rights reserved.
ServoCity™ and Actobotics™ are registered trademarks of Robotzone, LLC.

SAVE

Great educational discounts available. Apply today!

ENTER TO WIN!

Fun contests with awesome prizes - like \$1,000 worth of parts!



littleBits™
E L E C T R O N I C S

the **bitLab™**

USER-GENERATED HARDWARE IS HERE.

Get started at

littleBits.cc/bitLab

littleBits makes an expanding library of modular electronics that snap together with magnets. It's the easiest and most extensive way to learn and prototype with electronics. Create complex circuits in seconds: no soldering, wiring, or programming required (unless you want to).

The bitLab is the new marketplace for user-generated hardware. You can submit your own hardware designs for modules you want to see added to the littleBits library. If your design gets enough community support, we'll manufacture it, market it, sell it and give you a percentage of the sales!